1992

A computational study of torque and volumetric flow rates in a confined rotating fluid and optimum iteration parameters in cavity flow.

Edward. Lang
University of Windsor

National Library of Canada
Acquisitions and Bibliographic Services Branch
395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada
Direction des acquisitions et des services bibliographiques
395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Canada

A COMPUTATIONAL STUDY OF TORQUE AND VOLUMETRIC FLOW RATES IN A CONFINED ROTATING FLUID AND OPTIMUM ITERATION PARAMETERS IN CAVITY FLOW

by

Edward Lang

A Dissertation
Submitted to the Faculty of Graduate Studies and Research
Through the Department of Mechanical Engineering
in Partial Fulfilment
of the Requirements for the degree of
Doctor of Philosophy
at the University of Windsor

Windsor, Ontario, Canada

1992

ISBN 0-315-78918-2

ABSTRACT

The problem of laminar steady flow in a stationary cylinder with a rotating top disk was studied numerically. Three governing equations in cylindrical coordinates were solved by the alternating-direction implicit (ADI) method. To characterize the flow, three bulk quantities were selected, namely, the torque coefficient and the primary and secondary volumetric flow rates. Determination of the torque coefficient presented a difficulty because a singularity exists in the velocity gradient at the corner where the rotating disk and the stationary cylinder meet. This problem was overcome by specifying a gap between the disk and cylinder and incorporating this into the boundary conditions. The results obtained using these boundary conditions compared favourably with previous experimental, analytical and computational studies. The relevant parameters for the problem were the rotational Reynolds number, the aspect ratio (the ratio of the height of the cylinder to its radius) and the gap The ranges of parameters investigated were as follows: Reynolds number from 1 to 10^5; aspect ratio from 0.02 to 3; and gap size from 0.1% to 10% of the cylinder radius. The results indicated that the bulk quantities were dependent on the Reynolds number and the aspect ratio. The torque coefficient was also dependent on the gap, while the volumetric flow rates were only weakly dependent on the gap. For high aspect ratios, the bulk quantities approached constant values.

In addition, the effect of iteration parameters on convergence of the cavity problem was studied. The stream function equation was solved by the SOR (successive over-relaxation) method and the vorticity equation by the ADI method. The results obtained were contour plots of the number of iterations required for convergence in the iteration parameter space and graphs of the optimum iteration parameters as functions of Reynolds number and grid spacing. The range of values examined were from Re = 10 on a 21x21 grid to Re = 1000 on a 101x101 grid. It was found that the iteration parameter space contained four regions: a converging region, an overflow region (associated with numerical instability), an underflow region and a nonconverging region. The optimum iteration parameters were dependent on the Reynolds number and the grid spacing, and a strong coupling between iteration parameters was shown.

To my parents

ACKNOWLEDGEMENTS

The author wishes to express his gratitude to his supervisors, Drs. K. Sridhar and N.W. Wilson, whose support, guidance and patience made this work possible. The assistance of Dr. R.M. Barron is also gratefully acknowledged. The author took several graduate courses from him that proved very useful for this work. In addition, he has always been available when asked for assistance. Appreciation is also extended to Drs. W.T. Kierkus and G.W. Rankin who through questions and discussion have helped to keep this work focused.

The assistance rendered by the computer consultants is also gratefully acknowledged, especially, Mrs. J. Sreedharan. Thanks are also extended to Mr. W. Beck for technical assistance. Finally, it is with pleasure that the author acknowledges the contribution of Dr. K.R. Kumar. He gave freely of his time and ideas, and the many discussions with him proved most helpful.

This study was financially supported through a University of Windsor Postgraduate Scholarship and grants from the Natural Science and Engineering Research Council of Canada (Grant Number A-2190 and A-1403).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

$A, A^{\cdot},...,E^{\cdot}$	ADI coefficients
b	gap
C_m or Cm	torque coefficient
d	width of the rotating flow separation bubble
h	distance of the rotating flow separation bubble from the bottom of the cylinder
H	height of cylinder
I	iterations required for convergence
I_{opt}	iterations at optimum iteration parameters
L	characteristic length of cavity flow
M	number of grid points in x- or r-direction
N	number of grid points in y- or z-direction
Q_p or Qp	primary volumetric flow rate
Q_s or Qs	secondary volumetric flow rate
(r,θ,z)	cylindrical coordinates for rotating flow
$(r_{v.c.},z_{v.c.})$	position of the vortex centre in the physical domain
R_c	radius of cylinder
R_d	radius of disk
Re	Reynolds number, UL/ν or $\Omega R_c^2/\nu$
s	height of the rotating flow separation bubble

t	time
t_{opt}	CPU time at optimum iteration parameters
u	velocity in x-direction for cavity flow or in r-direction for rotating flow
U	characteristic velocity for cavity flow
v	velocity in y-direction for cavity flow or in θ-direction for rotating flow
w	velocity in z-direction for rotating flow
(x,y)	rectangular coordinates for cavity flow

Greek symbols

α,γ	stretching parameters
β	$\Delta x/\Delta y$ or $\Delta\xi/\Delta\eta$
δ	aspect ratio, H/R_c
Δ	difference in variable
Δt	ADI time step for cavity flow vorticity solver
Δt_{opt}	optimum ADI time step
ζ	vorticity for cavity and rotating flow
$\zeta_{v.c.}$	vorticity at the vortex centre
μ	dynamic viscosity
ν	kinematic viscosity
(ξ,η)	transformed (r,z) coordinates

ξ_d	radius of disk in computaticnal domain
$(\xi_{v.c.},\eta_{v.c.})$	position of the vortex centre in the computational domain
ρ	density
σ_v	iteration parameter for θ-velocity ADI solver, $\Delta\xi^2 Re/\Delta t$
σ_ψ	iteration parameter for stream function ADI solver, $\Delta\xi^2/\Delta t$
σ_ζ	iteration parameter for vorticity ADI solver, $\Delta\xi^2 Re/\Delta t$
τ	shear stress
ψ	stream function for cavity and rotating flow
$\psi_{v.c.}$	stream function at the vortex centre
ω	SOR iteration parameter
ω_{opt}	optimum SOR iteration parameter
Ω	angular velocity of rotating disk

Superscripts

'	dimensional variables
n	time step

Subscripts

i,j	x and y or r and z grid point
w	node point on wall boundary
opt	optimum variable

Special symbols

N	number of stream function contours

CHAPTER 1: INTRODUCTION

The subject of this dissertation is rotating fluid flow. Rotating fluid flow is important in many areas of engineering, such as rotating machinery, power transmission systems, chemical processing, computer storage devices, crystal growth processes and fluid flows in the environment.

The specific problem dealt with in this dissertation is shown in Figure 1.1. The fluid is confined in a cylinder of height H and radius R_c. The sidewall and endwall are stationary and the top is a rotating disk. The fundamental parameters of the flow are the rotational Reynolds number, Re, and the aspect ratio, δ, which is the ratio of the height of the cylinder to its radius. Another important parameter, whose significance will be discussed shortly, is the gap, b, between the outer edge of the rotating disk and the sidewall. The work is limited to laminar steady state flow without heat transfer.

The method used to study this flow is numerical. For a flow of this nature, using numerical techniques has some important advantages. First, it is easy to vary the Reynolds number, aspect ratio and gap in a computer program. To create an experiment with the same flexibility would be costly. Second, computational solutions give detailed velocity information over the whole flow field, from which volumetric flow rates can be calculated. To accomplish this experimentally requires a large number of measurements using intrusive devices like pitot tubes or hot wires, or complicated and expensive devices like

LDAs (laser-Doppler anemometers).

The main purpose of this research was to determine the effect of parameter variation on three bulk quantities: the torque coefficient, the primary volumetric flow rate and the secondary volumetric flow rate.

For practical purposes, the calculation of the torque required to rotate the top disk is of fundamental importance, but, unfortunately, this calculation is not straightforward. The problem arises from a discontinuity between the rotating disk and the sidewall. Points along the disk surface have a tangential velocity Ωr, but at the sidewall the velocity is zero; where they meet, a discontinuity exists, and the velocity gradient is singular here. Since the torque coefficient is a function of this gradient, it can not be calculated. In the past, most researchers who used computational methods did not present torque results. The few who did either ignored the singularity or proposed questionable methods to overcome it.

In fact, physically, a gap must exist between the rotating disk and the sidewall. In this research, the problem of the singularity is dealt with by introducing this gap into the boundary conditions. Now the torque is not only a function of the Reynolds number and the aspect ratio, but also of the size of the gap.

The other two bulk quantities are the primary and secondary volumetric flows. As shown in Figure 1.1, the flow field consists of two distinct components: a primary and a secondary flow. The primary flow is in the θ-

direction and is driven by the rotation of the disk. The secondary flow is in the r-z plane and is driven by centrifugal force. The total flow rates of these two flows are given by the primary and secondary volumetric flow rates. Information on the behaviour of these is important for problems like mixing.

In addition to the above work, a secondary problem was investigated: the determination of the optimum iteration parameter. The numerical method used to solve the governing equations was iterative, in which case an iteration parameter must be selected. The iteration parameter may be a time step or a factor used to accelerate convergence; in any case, it controls the rate of convergence. In fact, it governs not only the rate of convergence, but also whether the process converges or becomes numerically unstable and diverges. The question which arises is how can one select the iteration parameter that will give a solution in the fastest time: the optimum iteration parameter.

In computational fluid dynamics, the method commonly used to find the optimum iteration parameter is von Neumann stability analysis. In this method, a Fourier series expansion is made of the finite-difference equation, and each component of the expansion is considered separately to determine whether the computational method is stable or not. The optimum iteration parameter is considered to be the largest one permitted before instability occurs.

There are, however, several problems with this and other methods of stability analysis. The method is local, i.e., it is applied to each individual node in the computational domain. If each node is stable, then the solver is

considered stable; however, there is no rigorous proof that this is true. In addition, the effects of boundary conditions, nonlinearity and coupling between governing equations are ignored.

Due to these problems, a different approach was used: solve the problem at different parameter values and from these results determine the optimum iteration parameter. This part is the optimization study of the dissertation.

The problem with applying this technique to the rotating flow problem is that it has three governing equations, and, therefore, was too complicated for a preliminary investigation. The cavity problem was chosen instead, since it has only two governing equations.

The cavity problem is shown in Figure 1.2. The fluid is contained in a square two-dimensional cavity and is driven by a plate which slides along the top. The governing equations were solved by the SOR (successive over-relaxation) and the ADI (alternating direction implicit) methods.

It was hoped that the results of this study would provide information that could be applied to the rotating flow problem. During program development for rotating flow, it was found that convergence at high Reynolds number was poor. A number of procedures were tried to improve convergence, including changing the solver of the stream function equation from the SOR method to the ADI method--this was the one finally used. This made it impossible to directly apply the results of the optimization study to the rotating flow problem; however, the

results obtained were of interest and are, therefore, presented here.

The dissertation is organized as follows. Chapter 2 contains a survey of the literature. In Chapter 3 the governing equations are given and the solution methods are described. Results and discussion are in Chapter 4, and Chapter 5 contains conclusions and recommendations. Appendix A contains the coefficients of the finite-difference equations. Appendices B and C contain a listing of the computer programs for the two problems. Finally, Appendices D and E contain data from the results of the cavity and rotating flow problems, respectively.

CHAPTER 2: LITERATURE SURVEY

2.1 Cavity Flow and the Optimum Iteration Parameter

Cavity flow has always provided a standard test problem with which to investigate numerical methods, and, therefore, the literature in this area is large. Of interest for this work are the results of Benjamin [B2], Benjamin and Denny [B3], Ghia, et al. [G1] and Nallasamy and Prasad [N1] who have all done numerical studies of cavity flow at high Reynolds number. Benjamin [B2] provides a thorough review of cavity flow.

Several approaches are commonly used to evaluate the stability of the methods used in computational fluid dynamics. The literature in this area is massive, and a thorough discussion is beyond the scope of this dissertation, although a brief discussion will be given. The reader is also referred to Roache [R2] for an overview of stability methods and their associated problems.

The most common method of stability analysis is that attributed to the mathematician von Neumann, which was described briefly in the Introduction. It was first reported in the literature by O'Brien, et al. [O1]. The von Neumann method is usually applied to interior points in the computational domain, but an interesting application of the method was done by Trapp and Ramshaw [T3] who investigated the effect of boundary conditions on stability. In Hirt's stability analysis [H1], the finite-difference equations are expanded in a Taylor series and the stability characteristics of the resulting partial differential equations are

analyzed. Gustafsson [G3] developed the A-stability method which depends on a norm of the finite-difference solution satisfying certain criteria.

The numerical methods used, for this work, to solve the cavity problem are the SOR and the ADI methods. The SOR method was developed by Frankel [F1] and Young [Y1]. Frankel applied the method to linear partial differential equations and obtained an eigenfunction expansion of the error to determine the convergence rates. Young also applied the method to linear partial differential equations and showed how to select the best iteration parameter.

Some of the techniques developed to improve the convergence rate for the SOR method involve variation of the iteration parameter. Carré [C1] developed a method which gives updated estimates of the optimum iteration parameter as the iterations are done. Brazier [B5] used the von Neumann method to develop the nodal SOR method where the iteration parameter varies with the nodes.

The ADI method was developed by Peaceman and Rachford [P3] and Douglas [D4]. They applied the method to the heat flux equation and used von Neumann's method to investigate stability. Wachspress and Habetler [W2] and Wachspress [W1] developed a method to select the optimum iteration parameter and applied this to the diffusion equation. In addition, a method of varying the iteration parameter in a cycle was developed. Warming and Beam [W3] developed necessary and sufficient conditions for stability of the ADI

method and applied them to linear partial differential equations.

Benjamin [B2] and Benjamin and Denny [B3] used the ADI method to solve the cavity problem. They used a nonuniform grid and improved convergence by varying the iteration parameter with the grid nodes. The von Neumann method was used for their analysis.

The problem with this previous work is that no attempt has been made to find the optimum iteration parameter for a complete computational fluid dynamics problem. This dissertation will attempt to do this.

2.2 Rotating Flow

Flow resulting from a rotating disk can be divided into two categories: free disks and disk in a housing. The work of this dissertation deals with a disk in a housing; therefore, the literature survey on free disks will be kept to the essentials. In addition, only work dealing with laminar flow will be considered. Unless otherwise mentioned, if the disk is in a housing, the housing is stationary.

2.2.1 Free Disks

The first work on flow driven by a rotating disk was by von Kármán [K1]. He considered an infinite rotating disk in an semi-infinite fluid which is at rest at infinity. He applied a similarity solution to the full Navier-Stokes equations and obtained an approximate solution to the resulting ordinary differential equations.

This is the only known case where a similarity transformation can be applied to the full Navier-Stokes equations (the usual procedure is to simplify the equations first). He reported the fundamental motion for this type of flow: a steady inflow of the fluid towards the disk which is forced outward by centrifugal effects, i.e., the disk behaves like a centrifugal fan. He also was the first to obtain an analytical expression for the torque required to rotate the disk. Cochran [C2] solved the von Kármán equations by a more accurate method and gave a corrected value for the torque. Reference [K2] is an English translation of von Kármán's paper, and Schlichting [S1] gives an outline of his procedure.

Batchelor [B1] considered the case of two infinite parallel disks rotating at the same or different speeds. He did not solve the equations, but gave qualitative arguments regarding the flow field. He reported that the flow consists of separate boundary layers on the disks, with a central core that rotates with a constant angular velocity.

Stewartson [S4] considered the same problem as Batchelor. For the case of one rotating and one stationary disk, he obtained a solution to the similarity equations for low Reynolds numbers. He concluded that a boundary layer exists only at the rotating disk and that the core is stationary--a result radically different from that of Batchelor. His results imply no torque on the stationary disk, and he confirmed this using what he called crude experiments.

Other researchers also considered this problem. Rogers and Lance [R3] obtained a numerical solution of equations obtained by series expansion

methods, while Pearson [P4] obtained a numerical solution of the similarity equations. They found that the trend at small Reynolds number is misleading and that Batchelor was correct.

Mellor, et al. [M3] considered two infinite parallel disks, one rotating and one stationary; they obtained a numerical solution for the similarity equations. They concluded that at high Reynolds numbers multiple solutions exist, and they mapped three branches. They also did experiments on finite-radius disks and reported agreement with one of the branches. They did not obtain evidence of the physical existence of the other two branches.

A more detailed discussion of free rotating disks can be found in the review paper by Zandbergen and Dijkstra [Z1] and in the book by Greenspan [G2]. For a description of experimental and analytical results of torque on free rotating disks, see Theodorsen and Regier [T1].

2.2.2 Disk in a Housing

Schmieden [S2] considered a disk in a housing. He assumed creeping motion, i.e., he neglected inertial terms, and he obtained a Bessel function solution. He took into account the aspect ratio and the gap and he obtained the torque as a function of the aspect ratio, the gap and the Reynolds number.

Schultz-Grunow [S3] did an analytical study. He assumed that boundary layers exist on the disk and endwall, and that there is a core of fluid rotating at a constant angular velocity. He ignored the effect of the sidewall and used the

Kármán momentum integral method. He derived the value of the torque as a function of the Reynolds number.

Daily and Nece [D1] obtained both analytical and experimental results for the torque for aspect ratios from about 0.01 to 0.25 and Reynolds numbers from about 10^3 to 10^7. Their experimental setup had a constant gap between the disk and the housing, and their analytical study was similar to Schultz-Grunow, but they included a boundary layer on the sidewall. They gave empirical and analytical formulas for the torque.

Up to this point, theoretical studies were confined to boundary layer methods. This changed around 1970 with the increasing availability of modern computers. Rasmussen [R1] considered the case where the disk and housing are rotating, but the disk rotates at a slightly different angular velocity; this allowed him to linearize the Navier-Stokes equations. He presented finite-difference solutions for Reynolds numbers up to 200.

Tomlan and Hudson [T2] obtained computational solutions for the full Navier-Stokes equations. They took $\delta = 1$ and calculated solutions for Reynolds number up to 100. They also gave approximate solutions for very high Reynolds number. They did this by assuming that there are boundary layers on the disk, sidewall and endwall, with an inviscid core rotating at a constant angular velocity. Using numerical methods they solved the equations and gave results for $0 \leq \delta \leq 2.0$. Lehmkuhl and Hudson [L1] compared these results at high Reynolds number to the results of their experimental work. They reported

good agreement for $\delta = 1.0$. They also did some mass transfer experiments.

Pao [P1] did a numerical study of the full Navier-Stokes equations for $\delta = 1$ and Reynolds numbers up to 200. He also presented some transient solutions for Reynolds numbers up to 5000, but did not obtain the steady-state solution in these cases. He also did some flow visualization experiments and reported qualitative agreement with the computational results.

Pao [P2] obtained computational solutions for $\delta = 1$ and Reynolds numbers up to 400. He also presented analytical results for creeping motion in the form of a Bessel function expansion and reported that these results agree with the computational results at low Reynolds number. He did torque calculations and compared his results with those of von Kármán [K1], Schultz-Grunow [S3] and Daily and Nece [D1]. At Reynolds number of about 400 his results approach those of Daily and Nece. In general, however, his torque results showed poor agreement with previous work, particularly at lower Reynolds number.

Using computational methods, Bertelà and Gori [B4] obtained results for $\delta = 0.5$, 1 and 2 and $Re = 100$ and 1000. They gave the values of the torque and the primary and secondary volumetric flow rates at these values. At $Re = 100$ and $\delta = 1$, they reported acceptable agreement with Pao [P2] for the torque.

Lugt and Haussling [L5] obtained computational results for a rotating housing with the disk rotating at a different velocity. They took $\delta = 1$ with

Reynolds numbers up to 1000. They reported oscillations in the flow field for certain Reynolds numbers.

Duck [D5] did a numerical study with the disk and the endwall rotating in the same and different directions. He took $\delta = 1$ and did calculations for Reynolds numbers up to 1200.

A significant advance in the study of rotating flows was the discovery of vortex breakdown. Vortex breakdown occurs when a separation bubble of slow moving fluid forms on the axis of rotation. It occurs at aspect ratios greater than one. Vogel [V1] did an experimental flow visualization study of vortex breakdown. He found one separation region and presented a graph showing its occurrence as a function of δ and Re. His results covered aspect ratios in the range from 1.25 to 2.25 and Reynolds numbers from 1000 to 3000.

Ronnenberg [R4] used an LDA to study vortex breakdown. He took Reynolds numbers from 1000 to 2200 and aspects ratios greater then one and presented plots of the stream function contours.

Lugt and Haussling [L6] did a numerical study. For Re = 1350 and $\delta = 1.58$, they found vortex breakdown, thereby confirming Vogel's [V1] experimental work.

An extensive experimental study of vortex breakdown was done by Escudier [E1] who reported flow visualization results for Re = 1000 to 3500 and $\delta = 1.0$ to 3.5. He found cases of one, two and three breakdown regions. In addition, he found that for some aspect ratios the breakdown region oscillates

when the Reynolds number exceeds a certain value. He presented a map of the occurrence of breakdowns and oscillations with respect to the aspect ratio and Reynolds number.

Lugt and Abboud [L3] did a computational study designed to test Escudier's results. They got very good agreement, reporting results of one, two, and three breakdowns and oscillations for the same values of Reynolds number and aspect ratios reported by Escudier. They also varied the temperature on the boundaries.

Lopez [L2] also did a computational study designed to test Escudier's results. As with Lugt and Abboud, he reported good agreement with Escudier. Brown and Lopez [B8] presented a theory for the physical mechanism of vortex breakdown.

Now some examples of more complicated problems will be considered. Piesche [P5] considered the case of mass flow through a hole in the centre of the endwall. He did theoretical work, based on a series expansion, and also presented experimental results.

Daniels, et al. [D2] did an experimental study of two rotating disks with mass throughflow for small aspect ratios. They considered flat plates as well as turbine plates, and they gave results for Reynolds numbers up to 10^7.

Hudson and Eibeck [H2] obtained experimental results for multiple co-rotating disks. They included read/write arms to simulate magnetic computer disks.

This concludes the literature survey. One difficulty with the previous work is the lack of reliable computational solutions for the torque. While Pao [P2] and Bertelà and Gori [B4] did attempt the calculation, it will be shown that their methods and results are not valid. In addition, the results they presented were for very limited values of Reynolds numbers and aspect ratios.

Another problem with the above work is the limited results given for the primary and secondary volumetric flow rates. Only Bertelà and Gori give any results, and these are for very limited parameter values. This dissertation will attempt to solve the problem with the torque calculation and will present values of torque and flow rates for a wide range of Reynolds numbers and aspect ratios.

CHAPTER 3: FORMULATION OF THE PROBLEMS

In this chapter the equations for the cavity flow and the rotating flow problems will be given. In addition, the solution methods will be described.

3.1 Governing Equations for Cavity Flow

The two-dimensional square cavity flow problem is shown in Figure 1.2. The flow is driven by the top plate which moves to the right with a velocity U. The cavity has a height and width of L. In nondimensional form, the stream function and vorticity equations are

$$\frac{\partial^2\psi}{\partial x^2} + \frac{\partial^2\psi}{\partial y^2} = \zeta, \tag{3.1}$$

$$\frac{\partial\zeta}{\partial t} + u\frac{\partial\zeta}{\partial x} + v\frac{\partial\zeta}{\partial y} = \frac{1}{Re}\left[\frac{\partial^2\zeta}{\partial x^2} + \frac{\partial^2\zeta}{\partial y^2}\right], \tag{3.2}$$

where the Reynolds number is

$$Re = \frac{UL}{\nu}, \tag{3.3a}$$

the x-component of the velocity is

$$u = \frac{\partial\psi}{\partial y}, \tag{3.3b}$$

the y-component of the velocity is

$$v = -\frac{\partial \psi}{\partial x} \tag{3.3c}$$

and the vorticity is

$$\zeta = \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}. \tag{3.3d}$$

Using L and U as the characteristic length and velocity, the variables in the above equations have been nondimensionalized by

$$t = \frac{U}{L}t', \qquad x = \frac{x'}{L}, \qquad y = \frac{y'}{L},$$

$$u = \frac{u'}{U}, \qquad v = \frac{v'}{U},$$

$$\psi = \frac{\psi'}{LU}, \qquad \zeta = \frac{L}{U}\zeta',$$

where the prime denotes dimensional variables.

The boundary conditions are as follows:

for $0 \le x \le 1$ and $y = 0$, $\quad u = v = \psi = 0;$ (3.4a)

for $x = 1$ and $0 \le y < 1$, $\quad u = v = \psi = 0;$ (3.4b)

for $0 \le x \le 1$ and $y = 1$, $\quad v = \psi = 0,\ u = 1;$ (3.4c)

for $x = 0$ and $0 \le y < 1$, $\quad u = v = \psi = 0.$ (3.4d)

3.2 Method of Solving the Cavity Flow Equations

The equations for stream function and vorticity were solved by finite-difference methods. The computational domain, shown in Figure 3.1, was divided up into an MxN grid, where M and N are the number of nodes in the x- and y-directions, respectively. At each node, the continuous functions ψ, ζ, u and v were replaced by the discrete values ψ_{ij}^n, ζ_{ij}^n, u_{ij}^n and v_{ij}^n. The solution was obtained by an iterative method, i.e., the values at iteration level n were used to calculate the values at iteration level n + 1. This procedure was continued until a steady-state solution was reached.

The stream function equation (3.1) was solved using the SOR method (details can be found in Reference [R2]). The finite-difference equation is

$$\psi_{ij}^{n+1} = (1-\omega)\psi_{ij}^n + \frac{\omega}{2(1+\beta^2)}\left[\psi_{i+1,j}^n + \psi_{i-1,j}^{n+1} + \beta^2\psi_{i,j+1}^n + \beta^2\psi_{i,j-1}^{n+1} - \Delta x^2\zeta_{ij}^{n+1}\right], \tag{3.5}$$

where $i = 1,...,M$,

$j = 1,...,N$,

ψ_{ij}^n = stream function at grid point (i,j) and iteration level n,

$\Delta x = 1/(M-1)$,

$\Delta y = 1/(N-1)$,

$\beta = \Delta x/\Delta y$ and

ω = iteration parameter ($1 \le \omega < 2$).

The purpose of ω is to accelerate convergence.

The vorticity, equation (3.2), was solved using the ADI method (details can be found in Reference [R2]). The finite-difference equations are

$$A_{ij}^{*}\zeta_{i-1,j}^{*} + (1+B^{*})\zeta_{ij}^{*} + C_{ij}^{*}\zeta_{i+1,j}^{*} = -A_{ij}\zeta_{ij-1}^{n} + (1-B)\zeta_{ij}^{n} - C_{ij}\zeta_{ij+1}^{n}, \quad (3.6a)$$

$$A_{ij}\zeta_{ij-1}^{n+1} + (1+B)\zeta_{ij}^{n+1} + C_{ij}\zeta_{ij+1}^{n+1} = -A_{ij}^{*}\zeta_{i-1,j}^{*} + (1-B^{*})\zeta_{ij}^{*} - C_{ij}^{*}\zeta_{i+1,j}^{*}, \quad (3.6b)$$

where ζ_{ij}^{n} = vorticity at grid point (i,j) and iteration level n,

ζ_{ij}^{*} = vorticity at grid point (i,j) and the intermediate iteration level

and

ζ_{ij}^{n+1} = vorticity at grid point (i,j) and iteration level n+1.

The coefficients A_{ij}, A_{ij}^{*}, B, B^{*}, C_{ij} and C_{ij}^{*} are given in Appendix A.

The advantage of this method is that equations (3.6) are tridiagonal. These are easily inverted using the tridiagonal algorithm [R2], also known as the Thomas algorithm.

A fundamental problem with the stream function-vorticity formulation is the determination of the vorticity at a no-slip boundary. Roache [R2] gives

$$\zeta_{w}^{n} = \frac{2(\psi_{w+1}^{n}-\psi_{w}^{n})}{\Delta n^{2}} + O(\Delta n) \quad (3.7)$$

as the safest formula to use from the point of view of stability. The subscript w denotes a node on the wall, and Δn is the grid spacing normal to the wall. This equation is obtained by substituting equation (3.1) into a Taylor series expansion of the stream function. Using the same method used to derive equation (3.7), the following are the boundary conditions for vorticity:

for $0 < x < 1$ and $y = 0$ ($i = 2,...,M-1$ and $j = 1$),

$$\zeta_{i1}^{n} = \frac{2(\psi_{i2}^{n}-\psi_{i1}^{n})}{\Delta y^2}; \tag{3.8a}$$

for $x = 1$ and $0 < y < 1$ ($i = M$ and $j = 2,...,N-1$),

$$\zeta_{Mj}^{n} = \frac{2(\psi_{M-1j}^{n}-\psi_{Mj}^{n})}{\Delta x^2}; \tag{3.8b}$$

for $0 < x < 1$ and $y = 1$ ($i = 2,...,M-1$ and $j = N$),

$$\zeta_{i,N}^{n} = \frac{2(\psi_{i,N-1}^{n}-\psi_{i,N}^{n}+\Delta y)}{\Delta y^2}; \tag{3.8c}$$

for $x = 0$ and $0 < y < 1$ ($i = 1$ and $j = 2,...,N-1$),

$$\zeta_{1j}^{n} = \frac{2(\psi_{2j}^{n}-\psi_{1j}^{n})}{\Delta x^2}. \tag{3.8d}$$

Vorticity at the corners were taken as the average of adjacent points.

It is also necessary to calculate u_{ij}^{n+1} and v_{ij}^{n+1} at the interior points. This is done by a central differencing of equations (3.3):

$$u_{ij}^{n} = \frac{\psi_{i,j+1}^{n}-\psi_{i,j-1}^{n}}{2\Delta y}, \tag{3.9a}$$

$$v_{ij}^{n} = -\frac{\psi_{i+1,j}^{n}-\psi_{i-1,j}^{n}}{2\Delta x}. \tag{3.9b}$$

The following is the method of solution:

1) For initial conditions take $\psi_{ij}^0 = \zeta_{ij}^0 = u_{ij}^0 = v_{ij}^0 = 0$ for all interior points;

2) Solve for ζ_{ij}^{n+1} at all interior points using equations (3.6);

3) Solve for ψ_{ij}^{n+1} at all interior points using equations (3.5);

4) Calculate u_{ij}^{n+1} and v_{ij}^{n+1} at all interior points using equations (3.9) evaluated at n+1;

5) Calculate the vorticity on the boundaries using equations (3.8);

6) Test for convergence;

7) If the convergence criterion is not met, go to 2.

The convergence criterion used was

$$\max\left|\zeta_{ij}^{n+1} - \zeta_{ij}^{n}\right| \leq 10^{-4},$$

for all i,j. Vorticity was selected because preliminary numerical experiments showed that the vorticity converged slower than the stream function.

All computations were done by programming the above algorithm in FORTRAN using double precision on an IBM 4381 computer. The programs are given in Appendix B.

3.3 Governing Equations for Rotating Flow

The rotating flow problem is shown in Figure 1.1. The fluid is confined in a cylinder and is driven by the top disk which rotates with angular velocity Ω. The cylinder has a height H and a radius R_c.

The governing equations used to solve this problem are those given by Lugt and Abboud [L3]. Three equations are required to describe the flow. Given in the cylindrical coordinates (r,θ,z) shown in Figure 1.1, they are the axisymmetric stream function and vorticity equations for the secondary flow and a Navier-Stokes equation for the conservation of θ-momentum for the primary flow. The computational domain is shown in Fig. 3.1. It is assumed that the flow is axisymmetric, i.e., $\partial/\partial\theta \equiv 0$; as a consequence, the problem is symmetric about the line $r = 0$.

In nondimensional form, the stream function and the vorticity equations are

$$\frac{\partial \psi}{\partial t} = \delta^2 \frac{\partial^2 \psi}{\partial r^2} - \frac{\delta^2}{r}\frac{\partial \psi}{\partial r} + \frac{\partial^2 \psi}{\partial z^2} - r\zeta, \tag{3.10}$$

$$\frac{\partial \zeta}{\partial t} + \frac{\partial (u\zeta)}{\partial r} + \frac{\partial (w\zeta)}{\partial z} - 2\frac{v}{r}\frac{\partial v}{\partial z} = \frac{1}{Re}\left[\frac{\partial^2 \zeta}{\partial r^2} + \frac{1}{r}\frac{\partial \zeta}{\partial r} - \frac{1}{r^2}\zeta + \frac{1}{\delta^2}\frac{\partial^2 \zeta}{\partial z^2}\right], \tag{3.11}$$

and the equation for conservation of θ-momentum is

$$\frac{\partial v}{\partial t} + \frac{\partial (uv)}{\partial r} + \frac{\partial (wv)}{\partial z} + 2\frac{uv}{r} = \frac{1}{Re}\left[\frac{\partial^2 v}{\partial r^2} + \frac{1}{r}\frac{\partial v}{\partial r} - \frac{1}{r^2}v + \frac{1}{\delta^2}\frac{\partial^2 v}{\partial z^2}\right], \tag{3.12}$$

where the aspect ratio is

$$\delta = \frac{H}{R_c}, \tag{3.13a}$$

the rotational Reynolds number is

$$Re = \frac{\Omega R_c^2}{\nu}, \tag{3.13b}$$

the r-component of the velocity is

$$u = \frac{1}{r}\frac{\partial \psi}{\partial z}, \tag{3.13c}$$

the z-component of the velocity is

$$w = -\frac{1}{r}\frac{\partial \psi}{\partial r} \tag{3.13d}$$

and the vorticity is

$$\zeta = \frac{\partial u}{\partial z} - \delta^2 \frac{\partial w}{\partial r}. \tag{3.13e}$$

Note that the definitions for ψ, ζ, u, v and Re given here are different from the ones given in Sections 3.1 and 3.2 for the cavity problem. It will be clear from the context which definitions are being used.

Using R_c, H and Ω as the characteristic values, the variables in the above equations have been nondimensionalized by

$$t = \Omega t', \qquad r = \frac{r'}{R_c}, \qquad z = \frac{z'}{H},$$

$$u = \frac{u'}{\Omega R_c}, \qquad v = \frac{v'}{\Omega R_c}, \qquad w = \frac{w'}{\Omega R_c},$$

$$\psi = \frac{\psi'}{\Omega R_c^2 H}, \qquad \zeta = \frac{H}{\Omega R_c}\zeta',$$

where the superscript prime denotes dimensional variables.

An unusual feature of this choice of characteristic values is that different lengths are used: R_c in the r-direction and H in the z-direction. This method has the advantage of giving the 1x1 computational domain shown in Figure 3.2. It is also the reason that the aspect ratio, δ, occurs in the governing equations. The reasons for using R_c as the characteristic length in the r-direction will be discussed further in Section 4.3.3.

Another point of interest is the consequence of the assumption of axisymmetry, $\partial/\partial\theta \equiv 0$. The problem has three spatial dimensions r, θ and z and three dependent variables ψ, ζ and v; however, the computational domain is two dimensional. This is a consequence of the assumption of axisymmetry, since all variables are functions of r and z only, i.e., $\psi = \psi(r,z)$, $\zeta = \zeta(r,z)$ and $v = v(r,z)$. The governing equations can now be separated into a Navier-Stokes equation for the primary flow and stream function-vorticity equations for the

secondary flow. The term $-2(v/r)\partial v/\partial z$ in the vorticity equation (3.11) provides the driving force for the secondary flow.

A final point to be noted is that equation (3.10) has been written in transient form by including the term $\partial\psi/\partial t$. This method, sometimes called the method of false transients [M1,M2], allows greater computational efficiency.

The boundary conditions are as follows:

for $0 \le r \le 1$ and $z = 0$,	$u = v = w = \psi = 0;$	(3.14a)
for $r = 1$ and $0 \le z \le 1$,	$u = v = w = \psi = 0;$	(3.14b)
for $0 \le r < 1$ and $z = 1$,	$u = w = \psi = 0,\ v = r;$	(3.14c)
for $r = 0$ and $0 < z < 1$,	$u = v = \psi = \zeta = 0,\ \partial w/\partial r = 0.$	(3.14d)

These equations imply that v is discontinuous at r=1 and z=1. This will be discussed in Section 4.3.

A technique that proved useful for this work was to cluster grid points near the boundaries. This technique is commonly used to help resolve high velocity gradients at the boundaries. Clustering is achieved by using a stretching function; for this work, one given by Anderson, Tannehill, and Pletcher [A1] was used:

$$\xi = \alpha_r + (1-\alpha_r)\frac{\ln\left[\dfrac{\gamma_r+(2\alpha_r+1)r-2\alpha_r}{\gamma_r-(2\alpha_r+1)r+2\alpha_r}\right]}{\ln\left[\dfrac{\gamma_r+1}{\gamma_r-1}\right]}, \tag{3.15a}$$

$$\eta = \alpha_z + (1-\alpha_z)\frac{\ln\left[\dfrac{\gamma_z+(2\alpha_z+1)z-2\alpha_z}{\gamma_z-(2\alpha_z+1)z+2\alpha_z}\right]}{\ln\left[\dfrac{\gamma_z+1}{\gamma_z-1}\right]}. \tag{3.15b}$$

The parameters α and γ control the location and degree of grid refinement, respectively. For example, if $\alpha_r = 0$, the grid is refined near $r = 1$ only; if $\alpha_r = 0.5$, near $r = 0$ and $r = 1$. As γ increases, the degree of grid refinement decreases; the grid becomes uniform for $\gamma \to \infty$, while $\gamma = 1$ is a singular point which packs all the grid points onto the boundaries.

Applying equations (3.15) to equations (3.10), (3.11) and (3.12) gives the following transformed stream function and vorticity equations:

$$\frac{\partial\psi}{\partial t} = \delta^2\left(\frac{d\xi}{dr}\right)^2\frac{\partial^2\psi}{\partial\xi^2} + \delta^2\left[\frac{d^2\xi}{dr^2} - \frac{1}{r(\xi)}\frac{d\xi}{dr}\right]\frac{\partial\psi}{\partial\xi} + \left(\frac{d\eta}{dz}\right)^2\frac{\partial^2\psi}{\partial\eta^2}$$

$$+ \frac{d^2\eta}{dz^2}\frac{\partial\psi}{\partial\eta} - r(\xi)\zeta, \tag{3.16}$$

$$\frac{\partial \zeta}{\partial t} + \frac{d\xi}{dr}\frac{\partial (u\zeta)}{\partial \xi} + \frac{d\eta}{dz}\frac{\partial (w\zeta)}{\partial \eta} - \frac{2v}{r(\xi)}\frac{d\eta}{dz}\frac{\partial v}{\partial \eta}$$

$$= \frac{1}{Re}\left[\left(\frac{d\xi}{dr}\right)^2\frac{\partial^2 \zeta}{\partial \xi^2} + \left(\frac{d^2\xi}{dr^2} + \frac{1}{r(\xi)}\frac{d\xi}{dr}\right)\frac{\partial \zeta}{\partial \xi} - \frac{1}{r(\xi)^2}\zeta\right.$$

$$\left. + \frac{1}{\delta^2}\left(\frac{d\eta}{dz}\right)^2\frac{\partial^2 \zeta}{\partial \eta^2} + \frac{1}{\delta^2}\frac{d^2\eta}{dz^2}\frac{\partial \zeta}{\partial \eta}\right]. \qquad (3.17)$$

The conservation of θ-momentum becomes

$$\frac{\partial v}{\partial t} + \frac{d\xi}{dr}\frac{\partial (uv)}{\partial \xi} + \frac{d\eta}{dz}\frac{\partial (wv)}{\partial \eta} + 2\frac{uv}{r(\xi)}$$

$$= \frac{1}{Re}\left[\left(\frac{d\xi}{dr}\right)^2\frac{\partial^2 v}{\partial \xi^2} + \left(\frac{d^2\xi}{dr^2} + \frac{1}{r(\xi)}\frac{d\xi}{dr}\right)\frac{\partial v}{\partial \xi} - \frac{1}{r(\xi)^2}v\right.$$

$$\left. + \frac{1}{\delta^2}\left(\frac{d\eta}{dz}\right)^2\frac{\partial^2 v}{\partial \eta^2} + \frac{1}{\delta^2}\frac{d^2\eta}{dz^2}\frac{\partial v}{\partial \eta}\right]. \qquad (3.18)$$

The metrics, $d\xi/dr$, $d^2\xi/dr^2$, $d\eta/dz$, $d^2\eta/dz^2$, are obtained by differentiating equations (3.15) with respect to r and z.

In addition, equations (3.13c) and (3.13d) become

$$u = \frac{1}{r(\xi)}\frac{d\eta}{dz}\frac{\partial \psi}{\partial \eta}, \qquad (3.19a)$$

$$w = -\frac{1}{r(\xi)}\frac{d\xi}{dr}\frac{\partial \psi}{\partial \xi}. \qquad (3.19b)$$

3.4 Method of Solving the Rotating Flow Equations

The governing equations were solved by finite-difference methods. The computational domain, shown in Figure 3.2, was divided up into an MxN grid, where M and N are the number of nodes in the ξ- and η-directions, respectively. At each node, the continuous functions ψ, ζ, u, v and w were replaced by the discrete values ψ_{ij}^n, ζ_{ij}^n, u_{ij}^n, v_{ij}^n and w_{ij}^n. The solution was obtained by an iterative method, i.e., the values at iteration level n were used to calculate the values at iteration level n + 1. This procedure was continued until a steady state solution was reached.

All three equations were solved using the ADI method (details can be found in Reference [R2]). For the stream function equation (3.16), the finite-difference equations are

$$A_i^*\psi_{i-1,j}^* + (\sigma_\psi + B_i^*)\psi_{ij}^* + C_i^*\psi_{i+1,j}^* = -A_j\psi_{i,j-1}^n + (\sigma_\psi - B_j)\psi_{ij}^n - C_j\psi_{i,j+1}^n - \Delta\xi^2 r(\xi_i)\zeta_{ij}^n, \tag{3.20a}$$

$$A_j\psi_{i,j-1}^{n+1} + (\sigma_\psi + B_j)\psi_{ij}^{n+1} + C_j\psi_{i,j+1}^{n+1} = -A_i^*\psi_{i-1,j}^* + (\sigma_\psi - B_i^*)\psi_{ij}^* - C_i^*\psi_{i+1,j}^* - \Delta\xi^2 r(\xi_i)\zeta_{ij}^n. \tag{3.20b}$$

For the vorticity equation (3.17), the finite-difference equations are

$$A_{ij}^{*}\zeta_{i-1,j}^{*} + (\sigma_\zeta + B_i^{*})\zeta_{ij}^{*} + C_{ij}^{*}\zeta_{i+1,j}^{*} = -A_{ij}\zeta_{i,j-1}^{n} + (\sigma_\zeta - B_j)\zeta_{ij}^{n} - C_{ij}\zeta_{i,j+1}^{n}$$

$$+ \frac{E_j}{r(\xi_i)}\left(v_{i,j+1}^{n} - v_{i,j-1}^{n}\right)v_{ij}^{n}, \qquad (3.21a)$$

$$A_{ij}\zeta_{i,j-1}^{n+1} + (\sigma_\zeta + B_j)\zeta_{ij}^{n+1} + C_{ij}\zeta_{i,j+1}^{n+1} = -A_{ij}^{*}\zeta_{i-1,j}^{*} + (\sigma_\zeta - B_i^{*})\zeta_{ij}^{*} - C_{ij}^{*}\zeta_{i+1,j}^{*}$$

$$+ \frac{E_j}{r(\xi_i)}\left(v_{i,j+1}^{n} - v_{i,j-1}^{n}\right)v_{ij}^{n}. \qquad (3.21b)$$

And for v, equation (3.18), the finite-difference equations are

$$A_{ij}^{*}v_{i-1,j}^{*} + (\sigma_v + B_i^{*})v_{ij}^{*} + C_{ij}^{*}v_{i+1,j}^{*} = -A_{ij}v_{i,j-1}^{n} + (\sigma_v - B_j)v_{ij}^{n} - C_{ij}v_{i,j+1}^{n},$$

(3.22a)

$$A_{ij}v_{i,j-1}^{n+1} + (\sigma_v + B_j)v_{ij}^{n+1} + C_{ij}v_{i,j+1}^{n+1} = -A_{ij}^{*}v_{i-1,j}^{*} + (\sigma_v - B_i^{*})v_{ij}^{*} - C_{ij}^{*}v_{i+1,j}^{*}.$$

(3.22b)

In these equations,

$i = 1,...,M,$

$j = 1,...,N,$

F_{ij}^{n} = function at grid point (i,j) and iteration level n,

F_{ij}^{*} = function at grid point (i,j) and the intermediate iteration level,

F_{ij}^{n+1} = function at grid point (i,j) and iteration level n+1,

$\Delta\xi = 1/(M-1)$ and

$\Delta\eta = 1/(N-1)$.

F represents either ψ, ζ or v. The coefficients A, A*, B, B*, C and C* are given in Appendix A. These coefficients are local to their respective equations, i.e., they are not necessarily interchangeable; therefore, they are given separately for each equation.

As with cavity flow, there is a problem with finite-differencing vorticity on no-slip boundaries. Lugt and Haussling [L5] compared the results obtained using a first-order equation with a second-order equation for vorticity on a no-slip boundary. They reported that there was no advantage to using the second-order instead of the first-order equation, but they did decided to use the second-order one. For this work, the first-order equation was used. Using the same method used to derive equation (3.7), the following are the boundary conditions for vorticity:

for $0 < \xi < 1$ and $\eta = 0$ (i = 2,...,M-1 and j = 1),

$$\zeta_{i1}^{n} = \frac{2(\psi_{i2}^{n}-\psi_{i1}^{n})}{r(\xi_i)\Delta\eta^2}\left(\left.\frac{d\eta}{dz}\right|_{j=1}\right)^2; \tag{3.23a}$$

for $\xi = 1$ and $0 < \eta < 1$ (i = M and j = 2,...,N-1),

$$\zeta_{Mj}^{n} = \frac{2\delta^2(\psi_{M-1j}^{n}-\psi_{Mj}^{n})}{\Delta\xi^2}\left(\left.\frac{d\xi}{dr}\right|_{i=M}\right)^2; \tag{3.23b}$$

for $0 < \xi < 1$ and $\eta = 1$ (i = 2,...,M-1 and j = N),

$$\zeta_{i,N}^{n} = \frac{2(\psi_{i,N-1}^{n} - \psi_{i,N}^{n})}{r(\xi_i)\Delta\eta^2}\left(\left.\frac{d\eta}{dz}\right|_{j=N}\right)^2; \qquad (3.23c)$$

in addition, at the line of symmetry, $\xi = 0$ and $0 \le \eta \le 1$ ($i = 1$ and $j = 1,...,N$),

$$\zeta_{1j}^{n} = 0. \qquad (3.23d)$$

Vorticity at the corners on the right-hand side were taken as the average of adjacent points.

The final boundary condition is on the line of symmetry where $\partial w/\partial\xi = 0$. Using a first-order one-sided finite-differencing formula gives the following:

for $\xi = 0$ and $0 < \eta < 1$ ($i = 1$ and $j = 2,...,N-1$),

$$w_{1j}^{n} = w_{2j}^{n}. \qquad (3.24)$$

It is also necessary to calculate u_{ij}^{n} and v_{ij}^{n} at the interior points. This is done by a central differencing of equations (3.19):

$$u_{ij}^{n} = \frac{1}{2r(\xi_i)\Delta\eta}\left.\frac{d\eta}{dz}\right|_j\left(\psi_{i,j+1}^{n} - \psi_{i,j-1}^{n}\right), \qquad (3.25a)$$

$$w_{ij}^{n} = -\frac{1}{2r(\xi_i)\Delta\xi}\left.\frac{d\xi}{dr}\right|_i\left(\psi_{i+1,j}^{n} - \psi_{i-1,j}^{n}\right). \qquad (3.25b)$$

The following is the method of solution:

1) For initial conditions take either

   a) $\psi_{ij}^0 = \zeta_{ij}^0 = u_{ij}^0 = v_{ij}^0 = w_{ij}^0 = 0$ for all interior points, or

   b) the results of a previous calculation;

2) If the initial conditions of 1a are used, then equations (3.22) are used to create a initial distribution for the v-component of velocity. This was found, in preliminary calculations, to prevent underflow.

3) Solve for ζ_{ij}^{n+1} at all interior points using equations (3.21);

4) Solve for ψ_{ij}^{n+1} at all interior points using equations (3.20);

5) Calculate the vorticity on the boundaries using equations (3.23);

6) Solve for v_{ij}^{n+1} at all interior points using equations (3.22);

7) Calculate u_{ij}^{n+1} and w_{ij}^{n+1} using equations (3.25) evaluated at n+1 and w_{ij}^{n+1} using equation (3.24) evaluated at n+1;

8) Test for convergence;

9) If the convergence criterion is not met, go to 3.

The selection of the convergence criterion involved some differences from the cavity problem. Unlike cavity flow, it was found by numerical experiments that ψ converged slower than ζ. It was also found that the criterion used in the cavity flow was not stringent enough. Lugt and Abboud [L3] used

$$\sum_{ij} \left|\psi_{ij}^{n}-\psi_{ij}^{n-1}\right| < 10^{-4}.$$

Lopez [L2] was critical of their work, stating that they did not allow the process to reach a "true steady state"; Lopez himself did not state what convergence criterion he used. For this work,

$$\max\left|\frac{\psi_{ij}^{n}-\psi_{ij}^{n-1}}{\psi_{ij}^{n}}\right| \leq 10^{-5}$$

for all i,j and $\psi_{ij}^{n} \neq 0$ was used. If a less stringent criterion was used, the iteration process would sometimes stop prematurely.

The value of the right-hand side of the inequality was chosen by numerical experiments. A series of runs were done using different values. The largest one for which the solution did not vary in the third significant digit was used for the convergence criterion. This value was 10^{-5}.

All computations were done by programming the above algorithm in FORTRAN using double precision on an IBM 4381 computer. The programs are given in Appendix C.

3.5 Calculation of Bulk Characteristics for Rotating Flow

To characterize the flow, three bulk quantities were calculated. The first is the primary volumetric flow rate, Q_p, which gives the amount of fluid flow around the axis of symmetry. It is given by

$$Q_p = \delta \int_0^1 \int_0^1 v dr dz, \tag{3.26}$$

where Q_p was nondimensionalized using

$$Q_p = \frac{Q_p'}{\Omega R_c^3}. \tag{3.27}$$

Q_p' is the dimensional equivalent of equation (3.26).

The finite-difference form of the integration was obtained by weighting v over each element [R2]. This gives the second-order formula

$$Q_p \frac{1}{\delta \Delta r \Delta z} = \sum_{i=2}^{M-1} \sum_{j=2}^{N-1} V_{ij} + \frac{1}{2} \sum_{i=2}^{M-1} V_{i1} + \frac{1}{2} \sum_{j=2}^{N-1} V_{Mj} + \frac{1}{2} \sum_{i=2}^{M-1} V_{iN} + \frac{1}{2} \sum_{j=2}^{N-1} V_{1j}$$

$$+ \frac{1}{4}(V_{11} + V_{1N} + V_{MN} + V_{M1}). \tag{3.28}$$

If the stretching transformation, equation (3.15) is used, then

$$Q_p = \delta \int_0^1 \int_0^1 \frac{v}{(d\xi/dr)(d\eta/dz)} d\xi d\eta. \tag{3.29}$$

The second bulk quantity is the secondary volumetric flow rate, Q_s. It gives a measure of the secondary flow rate, and it is given by

$$Q_s = 2\pi\delta r_{v.c.}\int_0^{z_{v.c.}} u dz, \tag{3.30}$$

where the point $(r_{v.c.},z_{v.c.})$ is the vortex centre shown in Figure 3.2. Q_s was nondimensionalized using

$$Q_s = \frac{Q_s'}{\Omega R_c^3}. \tag{3.31}$$

Q_s' is the dimensional equivalent of equation (3.30).

If the stretching transformation, equation (3.15), is used, then

$$Q_s = 2\pi\delta\xi_{v.c.}\int_0^{\eta_{v.c.}} \frac{u}{d\eta/dz} d\eta. \tag{3.32}$$

Again, the point $(\xi_{v.c.},\eta_{v.c.})$ is the vortex centre. This equation was integrated using Simpson's rule.

A useful equation is given by the definition of the axisymmetric stream function [W4]

$$Q_s = 2\pi\delta\psi_{v.c.}, \tag{3.33}$$

where $\psi_{v.c.}$ is the value of the stream function at the vortex centre. This provides a check on the integration of equation (3.32).

Equation (3.30) gives only the flow rate for the central vortex; however, as will be shown later, there are some cases where one or more recirculation regions form in the secondary flow field. In these regions, however, the fluid is moving so slowly that its flow rate is usually several orders of magnitude less

that the main vortex. Therefore, the secondary volumetric flow rate was calculated only for the central vortex.

The third and final quantity is the torque coefficient, C_m, which gives the torque required to rotate the disk. To obtain the torque coefficient first consider the shear stress on the rotating disk:

$$\tau' = \mu \left. \frac{\partial v'}{\partial z'} \right|_{z'=H}.$$

An element of moment is obtained by multiplying this by an element of area and the moment arm. Now, integrating this element of moment along the disk gives the torque coefficient

$$C_m = \frac{8\pi}{\delta Re} \int_0^{R_d} \left. \frac{\partial v}{\partial z} \right|_{z=1} r^2 dr. \tag{3.34}$$

This was nondimensionalized using

$$C_m = \frac{2 \int_0^{R_d} dM'}{\frac{1}{2} \rho \Omega^2 R_c^5}, \tag{3.35}$$

where dM′ is the element of moment and R_d is the radius of the disk. The factor 2 in the numerator is used because C_m is usually defined for a disk submerged in a fluid; therefore, both sides provide drag force. However, the factor 2 is not always used, and care must be taken when comparing results from different sources.

If the stretching transformation, equation (3.15), is used, then

$$C_m = \frac{8\pi}{\delta Re}\left.\frac{d\eta}{dz}\right|_{z=1}\int_0^{\xi_d}\left.\frac{\partial v}{\partial \eta}\right|_{\eta=1}\frac{r(\xi)^2}{d\xi/dr}d\xi. \tag{3.36}$$

where ξ_d is the radius of the disk in the transformed coordinates. This equation was integrated using Simpson's rule.

To evaluate $\partial v/\partial \eta$ on the disk, a second-order one-sided finite-difference formula was used:

$$\left.\frac{\partial v}{\partial \eta}\right|_{i,N} = \frac{3v_{i,N}-4v_{i,N-1}+v_{i,N-2}}{2\Delta\eta}. \tag{3.37}$$

CHAPTER 4: RESULTS AND DISCUSSION

In this chapter the results of the research are presented. Section 4.1 is a discussion of the tests used to validate the computer programs. In Section 4.2, the results of the optimization study are presented. In Section 4.3, the difficulties with the calculation of the torque are discussed, along with the methods used to overcome these difficulties. The torque coefficient results are presented in Section 4.4, while Section 4.5 contains the primary and secondary flow rate results.

Among the results are some plots of stream function contours. Following the method of Lopez [L2], the contours are given by the formulas

$$\psi_i = \left(\frac{i}{N_{min}}\right)^3 \psi_{min}, \tag{4.1a}$$

$$\psi_j = \left(\frac{j}{N_{max}}\right)^3 \psi_{max}, \tag{4.1b}$$

where $i = 1,\ldots,N_{min}$ and $j = 1,\ldots,N_{max}$. N_{min} and N_{max} are the number of negative and positive contours, respectively. The values for ψ_{min}, ψ_{max}, N_{min} and N_{max} are given in Table 4.1 for the various figures.

4.1 Validation of the Programs

In this section, computational results from this study are compared to previous work to establish the validity of the methods and programs used. The cavity flow program given in Appendix B will be considered first, then the rotating flow program given in Appendix C.

4.1.1 Cavity Flow

Ghia, et al. [G1], using a multigrid technique, obtained solutions to the cavity problem for high Reynolds numbers and fine grids. In this section, results from the ADI-SOR solver, described in Section 3.2, will be compared with their work. Specifically, two Reynolds numbers have been selected: Re = 1000 and 3200. They used a 129x129 uniform grid, and the same was used here.

Figure 4.1 presents the stream function contours for Re = 1000. The most striking feature is the two recirculation regions in the bottom left and right corners. Ghia, et al. gave detailed information about the size of these regions, and Table 4.2 compares the results of this work with their's. In addition, values for ψ and ζ at the vortex centre, $\psi_{v.c.}$ and $\zeta_{v.c.}$, along with their coordinates are given. The following notation is used in Table 4.2 and illustrated in Figure 4.2: T for top, B for bottom, R for right, L for left, V for vertical and H for horizontal. For example, BRH means the maximum horizontal length of the bottom right recirculation region. From the table, it can be seen that the difference between the two set of values varies from 0% to about 2%.

In Figure 4.3, a comparison of u, the velocity component in the x-direction, is given along the line x = 0.5. The graph shows very good agreement between the two results. Maximum deviation, which occurs at the graph minimum, is about 2%.

In Figure 4.4, a comparison of v, the velocity component in the y-direction, is given along the line y = 0.5. The maximum deviations occur at the extrema and are, again, about 2%.

Now consider the case of Re = 3200. The stream function contours for this case are shown in Figure 4.5. There are two recirculation regions in the bottom corners and one near the top left corner. In Table 4.2, the relevant data for the regions and the vortex centre are given. It can be seen that the variation ranges from 0% to 10%, the biggest variation being for $\psi_{v.c.}$ and $\zeta_{v.c.}$.

Figure 4.6 is a comparison of u along the line x = 0.5. The deviation is between 2% and 12%, with the maximum at the relative extreme.

Figure 4.7 is a comparison of v along the line y = 0.5. The deviation is between 5% and 15%, with the maximum at the extrema.

Overall, the two methods show good agreement. At Reynolds number of 1000 the agreement is extremely good; at 3200, although not as good, it is still acceptable. For this optimization study, the highest Reynolds number used was 1000; therefore, it was concluded that the computational method of Section 3.2 is valid.

4.1.2 Rotating Flow

Escudier [E1] did an experimental flow visualization study of vortex breakdown in a cylinder with a rotating disk. Vortex breakdown occurs when, for certain Reynolds numbers and aspect ratios, recirculation regions form on the axis of symmetry. Lugt and Abboud [L3] and Lopez[L2] did computational studies of this phenomenon which confirmed the work of Escudier. It was felt that this experimental and computational work would provide a sensitive test of the validity of the computational method described in Section 3.4.

In Figure 4.8 the stream function contours for rotating flow with Re = 1994 and δ = 2.50 are given. This calculation was done on a 61x151 uniform grid, the same that Lopez used. Lugt and Abboud used either a 51x51 or an 81x81 grid with a stretching transformation to pack the grid near the boundaries.

This figure shows vortex breakdown with two recirculation regions on the line of symmetry. In Table 4.3, the dimensions of these regions, obtained from the different sources, are compared. The following notation is used in Tables 4.3 and 4.4 and illustrated in Figure 4.9: h is the height from the bottom of the cylinder to the bottom of the recirculation region, s is the height of the recirculation region itself and d is the width of the region. The subscript 1 refers to the lower bubble, while 2 refers to the top one. The values have been nondimensionalized by dividing by the cylinder radius or height, as appropriate.

Table 4.3 shows that there is good agreement between the various

results. The results of Lugt and Abboud show the largest deviation from the rest; this is probably due to the relatively coarse grid they used. It was not possible to obtain the values for the width from Escudier's results, because the photographs he published were only partial for the radius.

In Figure 4.10, the stream function contours for the case with Re = 2752 and δ = 3.25 are given. This calculation was done on a 61x196 uniform grid, the same that Lopez used. Again, Lugt and Abboud used either a 51x51 or an 81x81 nonuniform grid.

Again two recirculation regions have formed on the axis of symmetry, but this time the top one is elongated in the z-direction. Table 4.4 compares the dimension of the regions obtained from the various methods. Lopez did not give results for this case.

From the table, it can be seen that there is good agreement between the various methods. In most cases, however, the results of the present study are a little better than that of Lugt and Abboud's, relative to the experimental results. The one exception is s_2/H.

Overall, the computational method of Section 3.4 gives good agreement with the experimental and computational results of previous researchers. Since the phenomenon of vortex breakdown is complicated, it was concluded that the validity of the computational method has been proven.

4.2 Optimization Results

This section contains the results of the optimization study. The work is divided into two parts: a series of iteration contour graphs and graphs of the optimum iteration parameters.

The finite-difference equations for the cavity problem are given by equations (3.5) and (3.6). The rate of convergence of these equations is controlled by their iteration parameters: ω for the stream function equation and Δt for the vorticity equation. As these two values are varied, the number of iterations required for convergence, I, also varies. The purpose of this work is to determine the nature of this variation and the optimum iteration parameters, ω_{opt} and Δt_{opt}, i.e., the iteration parameters that produce convergence in the smallest number of iterations, I_{opt}. The CPU time for this case is t_{opt}.

Figures 4.12 to 4.23 show contours of I in the ω-Δt space for various combinations of Reynolds numbers and computational grids. All computational grids are uniform. The method used to produce these graphs was to cover the ω-Δt space with a mesh and then find the number of iterations required for convergence at each node. Each point on the graph now has three numbers associated with: ω, Δt and I. From this data the contour graphs were produced. The legend for these figures is given in Figure 4.11.

In Table 4.5, the values of ω, Δt and their increments, $\Delta\omega$ and $\Delta[\Delta t]$, are given with the CPU times in hours, t_{total}, used to produce each graph. In addition to the values given in Table 4.5, it was necessary, in some cases, to

add extra lines to resolve some regions. As an example of what was required, Figure 4.17 shows Figure 4.16 with the mesh.

It will be noted that some of the contour lines are not smooth. This is because, in an effort to save CPU time, a relatively coarse value of $\Delta\omega = 0.10$ was used in most cases. Even with this, the time required was large; for example, Figure 4.23 took 223 hours of CPU time. A smaller $\Delta\omega$ was, therefore, not considered feasible.

The iteration contour graphs were found to contain four regions: a converging region, an overflow region, an underflow region and a nonconverging region.

The region where the computational method converges is the converging region. The data in this region was normalized by dividing by I_{opt}. The contour lines were drawn from 1.00 to 3.00 in increments of 0.25. The point where the optimum parameters occurred is marked by the symbol shown in Figure 4.11; it has the normalized value 1.00. The values for Δt_{opt}, ω_{opt}, I_{opt} and t_{opt} are given in Table 4.6.

Next is the overflow region. Overflow indicates numerical instability; therefore, it is important to clearly mark this region. This region, however, presents some programming difficulties. With overflow, a program interrupt occurs followed by a standard corrective action: for IBM FORTRAN, the value of the register that overflowed is set to its largest possible floating-point value and then program execution continues. The number of these interrupts allowed

is unlimited. Since overflow indicates numerical instability, this will result in a nonterminating program; therefore, program execution was stopped after the first overflow. For IBM FORTRAN, subroutine ERRSET [I1] is available for this purpose (other systems may require a different method).

The underflow region is characterized by another type of interrupt. When underflow occurs, the standard corrective action is to set the register that underflowed to zero and continue execution. In this case also, the number of interrupts allowed is unlimited, and the subroutine ERRSET was used to terminate execution at the first underflow.

Underflow originates from the initial conditions. All initial conditions were set to zero so that all calculations had the same starting point--a fluid at rest. The fluid-at-rest initial conditions combined with very small Δt caused underflow. If initial conditions different from zero are used, underflow would not be a problem. This matter will be discussed further, later in this section.

Finally, there is the nonconverging region. When each run was made a limit on the number of iterations was set, and when this limit was reached, program execution was terminated. This defines the nonconverging region. It was found, by inspection, that the solution tended to oscillate back and forth without converging; however, this process may not continue indefinitely. For example, for Re = 10 on a 21x21 grid, it was found that some nonconverging points would slowly converge or slowly diverge, if the iteration process continued long enough. In general, then, it may be that many--if not all--points

in the nonconverging region are either slowly converging or slowly diverging.

The first graph in the series, Figure 4.12, is for Re = 10 with a computational grid of 21x21. More than half the graph is taken up by the overflow and nonconverging regions. The overflow region occupies the top part of the graph and has a relative minimum at about $\Delta t = 0.25$. The nonconverging region is located between the overflow and converging regions. The underflow region, although it exists, is too small to be seen in this case; it is a thin region parallel to the ω-axis near $\Delta t = 0$. The optimum point is near the top of the graph, between the overflow and underflow regions. This was the standard location for the optimum point. For this one case, the graph was extended to $\Delta t = 1.00$. All other graphs were extended only as far as required to close the contour line 3.00.

The converging region shows an important feature: the rate of convergence is dependent on a coupling between the two iteration parameters. If the rate of convergence was dependent only on Δt, then the contours would be straight lines parallel to the ω-axis; conversely, dependence only on ω would result in straight lines parallel to the Δt-axis. Although, roughly straight lines parallel to either axis appear in some areas of the graph, in general, this is not true. This is important, because when a stability analysis is done, the terms that couple the equations are taken as constant, i.e., the governing equations are treated as though their stability characteristics are independent. Figure 4.12 proves that this is not true. This dependency is even clearer in Figure 4.13.

Figure 4.13 is an expansion of the inset of Figure 4.12. This is not just a larger drawing of Figure 4.12, but, rather, the area has been redone with a finer mesh. This can be seen in Table 4.5. From this figure, the relationship of the optimum point to the surrounding regions, and the dependency of convergence on both Δt and ω is more clearly visible.

In Figures 4.14 to 4.19, for Re = 100, the computational grid varies from 36x36 to 101x101. As the computational grid gets finer the following occur: the underflow region increases; the nonconverging region decreases; the slope of the boundary of the overflow region increases; Δt_{opt} decreases by a factor of about 6.5; ω_{opt} increases by about 4%. By Figure 4.15, the underflow region has become distinct and is bounded on the right by an almost straight line. Most important of all, though, is that there is a significant decrease in the area of the converging region around the optimum point, and the optimum point itself is getting closer to the overflow region. In Figure 4.19, the optimum point is directly adjacent to the overflow region with no intervening contours.

Figures 4.20 and 4.21 give a series for Re = 250 with computational grids of 51x51 and 81x81. The same general trends described above are evident here. In this case, though, the optimum point is directly adjacent to the overflow region in both cases; therefore, this phenomenon occurs as the Reynolds number increases, the grid spacing decreases or both.

The Figures 4.18, 4.21, 4.22 and 4.23 are a series for a computational grid of 81x81 with Reynolds numbers of 100, 250, 500 and 750. As the

Reynolds number increases the following occur: the underflow region increases; the slope of the boundary of the overflow region decreases; Δt_{opt} increases by a factor of about 3.5; ω_{opt} increases by about 8%. The nonconverging region at first increases and then decreases.

Of the above, the most important results are the optimum points, since these are the points where the finite-differencing scheme converges in the fewest iterations. Therefore, determination of the convergence behaviour near the optimum point is important. Figures 4.24 to 4.26 give cross sections through the optimum points at $\Delta t = \Delta t_{opt}$ for Figures 4.13, 4.22 and 4.21, respectively. To help plot these graphs extra points were calculated.

Figure 4.24, for Re = 10 with a 21x21 computational grid, is the low Reynolds number and low grid density case. The graph shows that for $\omega < \omega_{opt}$, the variation in iterations is not great, but for $\omega > \omega_{opt}$, there is a dramatic increase in iterations as ω approaches the overflow region; therefore, the convergence rate is very sensitive to variation of ω near the overflow region. For example, at $\omega = 1.875$, it took 10 386 iterations to reach convergence, but at $\omega = 1.878$ the solution became unstable and overflow resulted. For this case, $\omega_{opt} = 1.72$ and $I_{opt} = 66$.

Figure 4.25, for Re = 500 with an 81x81 computational grid, is the high Reynolds number and high grid density case. For $\omega < \omega_{opt}$, the slope is greater than in Figure 4.24, while for $\omega > \omega_{opt}$, the increase in iterations as ω approaches the overflow region is sharper, but the relative increase in iterations

is not as great. For example, at ω = 1.959 88, I = 2319, but at ω = 1.959 89 instability occurs. In this case, I_{opt} = 886 at ω_{opt} = 1.95.

Iteration behaviour shows a significant difference in Figure 4.26 for Re = 250 with an 81x81 computational grid, which is the moderate Reynolds number and high grid density case. For $\omega < \omega_{opt}$, the slope of the curve is again much larger than in Figure 4.24, but for $\omega > \omega_{opt}$, there is only a small increase in I before overflow is reached. For example, at ω = 1.847 69, the solution converges in 907 iterations, but at ω = 1.847 70 overflow occurs. This is for a case where ω_{opt} = 1.84 and I_{opt} = 807. The consequence of this small increase in iterations followed by instability is seen in Figure 4.21 where the optimum point faces directly onto the overflow region without any intervening contour lines.

The above results show that the convergence rate is very sensitive to variation of ω when ω is near ω_{opt}. This sensitivity, as seen in Figures 4.24 and 4.25, was reported by Roach [R2]. He did not, however, report the significantly different behaviour of Figure 4.26, nor did he indicate how close ω_{opt} is to the overflow region; for example, for Figure 4.25, ω_{opt} = 1.95, but an increase of 0.01 in ω causes overflow.

Roache stated that the sensitivity of the convergence rate near ω_{opt} would make the determination of ω_{opt} tedious, but that the "experimental determination of an approximation to ω_o [ω_{opt}] is almost always worth while." At first it might appear that this behaviour does make the determination of ω_{opt} "tedious," but,

in fact, it can sometimes make it easier. The following procedure indicates how.

First an estimate must be made of Δt_{opt}. Now, starting with a value of ω known to be in the overflow region, decrease ω by steps of 0.05 until convergence is reached. This should give an approximation fairly close to the optimum. Of course, the opposite could be done: estimate ω_{opt} and, starting in the overflow region, decrease Δt until convergence; however, it will be shown later that the behaviour of Δt_{opt} is more predictable, and, therefore, is the better choice.

The success of this technique rests on two things: (1) the proximity of the optimum point to the overflow region, and (2) the small amount of CPU time used in the overflow region. The first point was established by the above results. For the second point, the use of the subroutine ERRSET, usually kept the CPU time small for overflow cases--in many cases under a minute. The main problem with the technique is if the nonconverging region is entered. It is, therefore, desirable to set a limit on the number of iterations allowed. As can be seen from Figure 4.12 to 4.23, though, the region in the immediate vicinity of the optimum point usually does not contain a nonconverging region.

The technique would be especially successful in cases like that shown in Figure 4.26, for a moderate Reynolds number with high grid density. In this case, the optimum point is very close to the overflow region and there is no dramatic increase in the convergence rate between the optimum point and the overflow region. For cases like Figures 4.24 and 4.25, the technique would not

be as effective, but should still prove useful.

The above technique requires an estimate of Δt_{opt}, and this was the purpose of the second part of the optimization study--the graphs of the optimum iteration parameters. From the first part, some optimum iteration parameters are already available (see Table 4.6). For the second part, a more extensive search was made.

The method used was to cover the ω-Δt space with a coarse mesh to get a rough estimate of where the minimum occurs. With this information, different values of Δt and ω were tried until a point was found where all the adjacent points had higher convergence rates or were in the overflow or underflow regions. This point was taken as the minimum. The search was done to three decimal places for Δt_{opt} and two for ω_{opt}. Of course, the problem with this method is that it is never possible to be sure the minimum has been found.

This part of the optimization study was done for Reynolds numbers of 10, 100, 250, 500, 750 and 1000 with computational grids of 21x21, 36x36, 51x51, 66x66, 81x81 and 101x101. All the data are given in Appendix D. In Figures 4.26 to 4.28 are plotted Δt_{opt}, ω_{opt} and I_{opt} versus N, the number of grid points in the x-direction.

When the data were plotted, it was found that some points did not fit the general trends. This occurred when the Reynolds number was high and the grid density low. The computational results for these cases were found to have poor accuracy. It was decided not to plot these data, although, they are

included in Appendix D.

Figure 4.27 gives the results for Δt_{opt}. The curves are well behaved with two trends apparent: for a fixed Reynolds number, Δt_{opt} decreases as the grid increases and for a fixed grid, Δt_{opt} increase as the Reynolds number increases.

Figure 4.28 gives the results for ω_{opt}. In this case, the curves are not as well behaved. For low Reynolds numbers, ω_{opt} first increases and then levels off as the grid increases. For a fixed grid, ω_{opt} decreases as the Reynolds number increases from 10 to 100; however, at Re = 250, ω_{opt} is between the curves for Re = 10 and 100.

For high Reynolds numbers with high grid densities an unusual phenomenon occurs. There is a significant drop in ω_{opt} at N = 101 for the Reynolds numbers 500, 750 and 1000. The reason can be seen by comparing Figures 4.20 and 4.21, the cases of Re = 250 with grids of 51x51 and 81x81. As the grid density increases, Δt_{opt} decreases, ω_{opt} increases and the size of the underflow region increases. As a result, the optimum point is getting closer to the underflow region. This trend continues in Figure 4.23, the case for Re = 750 with an 81x81 grid. Eventually, the optimum point will reach the underflow region. The dip in Figure 4.28 indicates that this has happened. This is confirmed by the data. For example, for the case of Re = 750 with a grid of 101x101, the optimum point occurs at Δt_{opt} = 0.063 and ω_{opt} = 1.80; at Δt = 0.062 underflow occurs for all ω; at Δt = 0.063 and ω = 1.81 overflow occurs. The optimum point has been bounded by the overflow and underflow

regions.

For some cases of high Reynolds numbers with high grid densities, it was found that this trend continued to the point where the underflow and overflow regions appeared to have joined--no converging region was found. Roache [R2] states that the selection of initial conditions is unimportant. The above results show that this is not true, since the problem with underflow can be avoided by not using the fluid-at-rest boundary conditions.

The data for Δt_{opt} does not show the same degree of variation because Δt_{opt} is less sensitive. The slope of the overflow region is large, therefore, a small variation of Δt_{opt} results in a large one for ω_{opt}. At the point for Re = 1000 and N = 101, however, the effect has become visible for Δt_{opt}. At this point, Δt_{opt} is slightly higher than it would be if the general trends of the curves are followed.

Figure 4.29 gives the resuits for I_{opt}. These results will help avoid problems with the nonconverging region by giving an estimate of the maximum number of iterations allowed. The graphs are well behaved with two trends: for a fixed Reynolds number, I_{opt} increases as the grid increases and for a fixed grid, I_{opt} increase as the Reynolds number increases.

4.3 Calculation of Torque Coefficient

Now the work on rotating flow will be presented. In this section, the problems associated with calculating the torque coefficient along with a method to overcome these problems will be discussed.

When the torque coefficient is calculated using equation (3.36), a problem arises. It can be seen from equations (3.14b) and (3.14c) that v is discontinuous at r = 1 and z = 1: on the rotating disk, v = r for r < 1, but on the sidewall, v = 0 for r = 1. A consequence of this is that $\partial v/\partial z$ is singular at this corner point. Lugt and Abboud [L3] gave an asymptotic form for the singularity:

$$\lim_{r\to 1} \frac{\partial v}{\partial z} \sim \frac{1}{1-r}. \tag{4.2}$$

Since the singularity is at a corner point and there are no mixed derivatives in the equations, the finite-difference equations do not involve the singularity. This is the reason that computational studies of rotating flow which did not deal with the torque could ignore this problem. It cannot be ignored if the torque is to be calculated.

4.3.1 Pao's Method

Obviously, to calculate the torque coefficient, a method to deal with the singularity is required. Pao [P2] assumed that there is a constant boundary layer on the disk near the tip. Specifically, he assumed $(1/r)\partial v/\partial z$ on the disk was constant near the tip. This gave him the formula

$$C_m = \frac{8\pi}{\delta Re}\left[\int_0^{r_0} \left.\frac{\partial v}{\partial z}\right|_{z=1} r^2 dr + \left(r^{-1}\left.\frac{\partial v}{\partial r}\right|_{z=1}\right)_{r=r_0} \int_{r_0}^{1} r^3 dr\right]. \qquad (4.3)$$

Using this formula, he did a series of computations for Reynolds number from 1 to 400 on a 41x41 uniform grid with $\delta = 1$. He assumed that the constant boundary layer extended across the last grid spacing, and, therefore, took $r_0 = 0.975$ which is one node from the corner. The effectiveness of this method is assessed below.

In this study, Pao's method was used to calculate the torque coefficient with $\delta = 1$ and Re = 1, 10, 100, 500 and 1000. Three different uniform grids were used: 21x21, 41x41 and 81x81. The results are given in Figure 4.30. For the 41x41 grid, the same Pao used, the results are the same as he published [P2]; however, across the grids, the maximum deviation is about 50%. Pao's method, therefore, is grid dependent.

To understand why this must be the case consider Figure 4.31, which shows the values of $\partial v/\partial z$ along the rotating disk for Re = 1 (the torque coefficient is dependent on $\partial v/\partial z$). As the grid spacing gets smaller, the gradient $\partial v/\partial z$ near the edge gets significantly larger--this is the effect of the singularity given in equation (4.2). Intuitively, this behaviour is obvious: as the grid spacing gets smaller, the velocity near the edge must go from v = r on the disk to v = 0 on the sidewall in an increasingly smaller distance. This is the reason for the grid sensitivity of Pao's method.

Another attempt to calculate the torque from computational results was made by Bertelà and Gori [B4]. They gave the formula

$$C_m = \frac{8\pi}{\delta Re}\int_0^1 \left|\frac{\partial v}{\partial z}\right|_{z=1} r^2 dr. \tag{4.4}$$

They used a 21x21 grid and gave C_m for δ = 0.5, 1 and 2 with Re = 100 and 1000. They did not explain how they dealt with the singularity and did not even indicate that they were aware of the problem. It was not possible to reproduce their torque results.

4.3.2 Boundary Conditions

The singularity results from assuming that there is no gap between the rotating disk and the sidewall, but in any physical application there must be a gap. The problem with the singularity, therefore, can be avoided by including the gap as part of the boundary conditions.

In this case, the boundary at the top of the computational domain, shown in Figure 3.2, is divided into two parts. The first section is occupied by the rotating disk, and the boundary conditions are the same as equation (3.14c):

$$\text{for } 0 \le r \le 1\text{-b and } z = 1, \qquad u = w = \psi = 0,\ v = r; \tag{4.5}$$

where $b = b'/R_c$ and b' is the length of the gap, i.e., $R_c = R_d + b'$. R_d is the radius of the disk.

It is now necessary to provide appropriate boundary conditions in the gap. One possibility is to assume symmetry. This would be true if the rotating disk were suspended midway in the cylinder. In this case, the boundary conditions are as follows:

$$\text{for } 1\text{-}b < r < 1 \text{ and } z = 1, \qquad w = \psi = \zeta = \partial u/\partial z = \partial v/\partial z = 0. \tag{4.6}$$

These are the symmetry boundary conditions. The condition $\zeta = 0$ can be derived by substituting equation (4.6) into equation (3.13e). For simplicity, it was assumed that the thickness of the disk is zero; otherwise, the annular space between the disk and the sidewall would have to be included as a step in the computational domain.

For the variables u and v, in equation (4.6), finite-difference equations are needed. Using a one-sided second-order formula, the following are obtained for u and v in the gap:

$$u_{iM} = \frac{4}{3}u_{i,M-1} - \frac{1}{3}u_{i,M-2}, \tag{4.7}$$

$$v_{iM} = \frac{4}{3}v_{i,M-1} - \frac{1}{3}v_{i,M-2}. \tag{4.8}$$

As with Pao's method, the symmetry boundary conditions were used to calculate the torque coefficient with $\delta = 1$ and Re = 1, 10, 100, 500 and 1000 on three different grids: 21x21, 41x41 and 81x81. The results are given in

Figure 4.32. The maximum deviation across the grids is about 10%--a significant improvement. The one exception to this is at Re = 1000 where it is about 15% This is because the data from the 21x21 grid at Re = 1000 is less accurate.

A problem with the above method is that if the disk is not at the midway point of the cylinder, the flow field above the disk is different from that below, and the symmetry boundary conditions are not valid. An alternate set of boundary conditions, therefore, would be useful.

In any practical application, the rotating disk would have a finite thickness. Since the disk is rotating concentric with the sidewall, an approximation of this would be the flow between two infinite concentric rotating cylinders. This type of flow is cylindrical Couette flow and is given in Schlichting [S1]. The problem dealt with in this dissertation reduces to the special case of an inner cylinder with an angular velocity Ω and a stationary outer cylinder, separated by a distance b. The velocity is then given by

$$v_c = v(r) = \frac{(1-b)^2}{1-(1-b^2)^2}\left[\frac{1}{r} - r\right]. \tag{4.9}$$

Now, the boundary conditions for the gap are as follows:

for 1-b < r < 1 and z = 1, $\quad u = w = \psi = 0, \; v = v_c.$ (4.10)

These are the Couette flow boundary conditions. The conditions u = w = 0 are

necessary for the flow to be consistent with the assumption of cylindrical Couette flow. Also, since u = w = 0, the region in the gap has, effectively, a no-slip boundary condition, and therefore equation (3.23c) is the correct form for the vorticity. Again, for simplicity, it was assumed that the thickness of the disk is zero.

Figure 4.33 gives the results of the torque coefficient using the Couette flow boundary conditions for the same combinations of aspect ratio, Reynolds numbers and grid spacings used in the other two cases. The maximum deviation is 5% for most Reynolds numbers except at 1000, were it is 15%; again, this is a consequence of using a 21x21 grid for such a high Reynolds number.

It, therefore, can be concluded that the symmetry boundary conditions and the Couette flow boundary conditions give consistent results and are not grid dependent.

An important question, though, is how do the methods compare with each other. Figure 4.34 compares them on an 81x81 grid. The maximum deviation occurs at low Reynolds numbers and is about 10%. At high Reynolds number, the variation drops to 1.5%. Therefore, it is concluded that the essential requirement necessary to calculate the torque coefficient is to include the gap in the boundary conditions in some consistent manner.

The close agreement between the two boundary conditions leads to the question of which one to use. It was not possible to do all calculations for both

boundary conditions, because of constraints of time and computer resources. It was decided to use only the Couette boundary conditions, because they are more general (they do not assume the disk is located midway in the cylinder) and because they converged slightly faster. When computational results were compared to the work of other researchers, though, the symmetry conditions were used when appropriate.

There was one significant difference between the boundary conditions. For the Couette boundary conditions, a recirculation region sometimes formed in the upper right-hand corner. Figure 4.35 shows this for $Re = 10^4$ and $\delta = 0.25$. Figure 4.36, plotted for the same parameters, used the symmetry boundary conditions but does not show the recirculation region.

The recirculation region increases in size as the Reynolds number increases. With respect to b, it decreases as b decreases and does not occur at the lower values of b. It does not occur at the very low aspect ratios nor the high ones: it first appears at $\delta = 0.10$ (with $Re = 5000$ and $b = 0.10$), at $\delta = 1.00$ it is very small and at $\delta = 1.5$ it has disappeared.

At this point, it is convenient to mention one other recirculation phenomenon that occurred. A small recirculation region in the lower right-hand corner is a standard feature for this type of flow. This can be seen in Figures 4.8 and 4.10. For high aspect ratios and low Reynolds number, however, this region becomes very large. Figure 4.37 is an example with $\delta = 3.00$ and $Re = 100$. At $\delta = 3.00$ and $Re = 500$, the recirculation region is

about 10% the size at Re = 100, while for $\delta = 2.50$ and Re = 100, it is about 25% the size. This phenomenon is not effected by b.

4.3.3 Selection of the Characteristic Length

It has been shown that in order to calculate the torque coefficient, it is necessary to specify appropriate boundary conditions in the gap between the rotating disk and the sidewall. As a result, there are two fundamental lengths in the r-direction: R_d, the radius of the disk and R_c, the radius of the cylinder. The problem is which one should be used as the characteristic length.

First note that R_c is related to R_d by

$$R_c = R_d + b'. \tag{4.11}$$

Also we have the following definitions:

$$Re_d = \frac{\Omega R_d^2}{\nu}, \tag{4.12a}$$

$$Re_c = Re = \frac{\Omega R_c^2}{\nu}, \tag{4.12b}$$

$$\delta_d = \frac{H}{R_d}, \tag{4.13a}$$

$$\delta_c = \delta = \frac{H}{R_c}, \tag{4.13b}$$

$$C_{m_d} = \frac{2M'}{\frac{1}{2}\rho\Omega^2 R_d^5}, \tag{4.14a}$$

$$C_{m_c} = C_m = \frac{2M'}{\frac{1}{2}\rho\Omega^2 R_c^5}. \tag{4.14b}$$

Using equation (4.11) it can be shown that these equations are related by

$$Re_c = \frac{Re_d}{(1-b)^2}, \tag{4.15a}$$

$$\delta_c = (1-b)\delta_d, \tag{4.15b}$$

$$C_{m_c} = (1-b)^5 C_{m_d}, \tag{4.15c}$$

where $b = b'/R_c$.

Let R^* represent the characteristic length. Three possible choices for R^* are considered.

For the first method take $R^* = R_c$. In this case, $Re = Re_c$ and $\delta = \delta_c$. In the literature, this method is used for computational studies, analytical studies that model the gap and experimental studies that investigate the flow phenomenon but don't consider the torque. It has the advantages of normalizing the r-coordinate to one, so that, the computational domain is 1x1. It also introduces the gap length, b, into the formulas which can be useful for comparison purposes. This point will be illustrated later. The disadvantage of

this method is that it is the disk that is rotating and not the cylinder, so, it would be more natural to choose R_d for R^*.

For the second method take $R^* = R_d$. In this case, $Re = Re_d$ and $\delta = \delta_d$. In the literature, this method is used for analytical and experimental studies that investigate the torque. The problem of applying this method to a computational study is that the computational domain becomes $(1+b'/R_d)x1$. For b' arbitrary, this boundary can result in difficulties. A way of avoiding this is given by the third method.

For the third method take $R^* = R_c$. But in this case take $Re = Re_d$ and $\delta = \delta_c$. This method is the result of normalizing everything using R_c--giving a 1x1 computational domain--but defining the Reynolds number in terms of the disk radius. In this case, the Reynolds number in governing equations (3.11) and (3.12) must be replaced by

$$\frac{Re_d}{(1-b)^2}$$

The disadvantage of this method is that it can be confusing, especially when comparing results from different sources.

Any of these methods could be used, but the first was selected because it is the simplest. Different researchers use different methods, however, and care should be taken when comparing results.

4.3.4 Comparison with Experimental and Analytical Results

In this Section, the method for calculating the torque coefficient described above will be compared to experimental and analytical results available in the literature.

Daily and Nece [D1] investigated the torque on a disk rotating in a housing by analytical and experimental methods. Experimentally, they dealt with small aspect ratios, δ_d = 0.0127 to 0.217, and they varied the Reynolds number from about 10^3 to 10^7. The disk and housing radii were constant for all their experiments and $b = b'/R_c$ = 0.006 33.

For laminar flow, they identified two regimes. In Regime I, the boundary layers on the top rotating disk and the stationary endwall are merged, the velocity profile is linear and there is no secondary flow. Using these assumptions, they gave the theoretical equation for the torque coefficient as

$$C_{m_s} = \frac{2\pi}{\delta_d Re_d}. \tag{4.16}$$

Comparing this with their experimental results, they concluded that it was also the best empirical equation.

In Regime II, the boundary layers are separated, and between them a core of fluid rotates with a constant angular velocity. Based on their experiments, they gave the empirical equation

$$C_{m_d} = \frac{3.70\delta_d^{1/10}}{Re_d^{1/2}}. \qquad (4.17)$$

Using equations (4.15), equations (4.16) and (4.17) become

$$C_m = \frac{2\pi(1-b)^4}{\delta Re}, \qquad (4.18)$$

$$C_m = \frac{3.70\delta^{1/10}(1-b)^{39/40}}{Re^{1/2}}. \qquad (4.19)$$

The presence of b in these equations is due to the fact that the Reynolds number is based on R_c, but the integral limit of equation (3.36) is R_d. The factors with b, though, do not model the end effect resulting from the interaction of the edge of the disk with the sidewall.

In Figure 4.38, equation (4.18) and (4.19) are compared with the computational results for b = 0.0063. The calculations were done on a 601x13 grid using the stretching transformation given by equations (3.15)--without stretching, it would be impossible to place a grid point on the edge of the disk, since b = 0.0063. The symmetry boundary conditions were used because of their experimental setup. The calculations were done for δ = 0.02 and Re = 1 to 10^6, although, Daily and Nece presented experimental results only as low as 10^3. Overall, excellent agreement is obtained between experiment and computation; however, there is some deviation around Re = 20 000 and 100 000. At Re = 20 000, there is a transition from Regime I to II. Experimentally,

Daily and Nece found that there is a smooth curve from one regime to the other, as was found in the computational results. The same phenomenon occurs around Re = 100 000; this time, there is a transition to a turbulent regime.

In Figure 4.39 the same comparison is done for $\delta = 0.10$ and $b = 0.0063$. These calculations were done on a 601x61 nonuniform grid. For Re > 1000, there is very good agreement with the experimental results, while for Re ≤ 1000, there is significant variation. The two lowest points on Figure 4.39 are for Re = 500 and 750. The lowest Reynolds number for which Daily and Nece gave results was about 700, i.e., they gave no experimental points in Regime I. It was impossible to obtain a converging solution for Re = 100 000. Daily and Nece indicate that this point is near a turbulent region.

It is seen that the assumptions of Regime I give good results for $\delta = 0.02$ but not for 0.10. To understand why this is so, it is necessary to examine the velocity profiles and see if they are linear. Figure 4.40 gives the velocity profiles of v for Re = 100 and $\delta = 0.02$ at r = 0.5 and at the tip of the rotating disk, r = 0.9937. According to Daily and Nece, these parameter should place the flow in Regime I, and the profiles should be linear. It is linear at r = 0.5 but not at the tip, because of the influence of the sidewall. The torque, though, is dependent on $\partial v/\partial z$ on the disk, i.e., the slope of the curves in Figure 4.40 at z = 1. For r = 0.9937, the slope is not very different from what it would be if the profile was linear. This is why Figure 4.38 shows such good agreement at

Re = 100--the influence of the sidewall is not important in this case and the assumption of linearity is a good approximation.

The situation is different in Figure 4.41 which gives the velocity profile for Re = 500 and $\delta = 0.10$. These parameters should place the flow at the edge of Regime I. At r = 0.5 the profile is almost linear, but at the tip it is not. This time, $\partial v/\partial z$ is much greater than it would be if it was linear. In this case, the influence of the sidewall is significant, and this is reflected in the deviation shown in Figure 4.39.

The reason for the different behaviour at $\delta = 0.02$ and $\delta = 0.10$ is, probably, because of the influence of the endwall. For very small δ, the boundary layer on the rotating disk is strongly influenced by the endwall, and the effect of the sidewall is negligible. As δ increases, the endwall influence decreases, and the effect of the sidewall increases. Therefore, the assumption of linearity of Regime I is only good for $\delta \ll 1$ and, more specifically, for $\delta < 0.10$.

Since the assumptions of Regime I are not always met, it is interesting to determine what the situation is for the assumptions of Regime II; namely, two boundary layers with a central core rotating at constant angular velocity. Figures 4.42 and 4.43 show the velocity profiles for Re = 30 000 with $\delta = 0.02$ and Re = 10 000 with $\delta = 0.10$, respectively. These values place the flow at about the midpoint of the Regime II lines of Figures 4.38 and 4.39.

Figure 4.42 , for $\delta = 0.02$, shows that the assumptions of Regime II are

not satisfied. This is probably because the distance between the rotating disk and the endwall is so small that a central core cannot form. Figure 4.43, for $\delta = 0.10$, shows that at $r = 0.5$ the assumptions of Regime II are satisfied, but at $r = 0.9937$ they are not.

It appears that the assumption of only two regimes is a simplification and that the actual flow patterns are more complicated. This is an area that requires further study. It should also be noted that, due to the limitation of methods available, Daily and Nece did not experimentally measure the velocity profiles for laminar flow, only for turbulent flow.

The only analytical study that attempted to deal with the gap was done by Schmieden [S2]. He considered the case for very small Reynolds number and used the creeping flow approximation, i.e., inertial terms were neglected. He also assumed zero pressure gradient and used the symmetry boundary conditions in the gap. He obtained a Bessel function expansion of the solution and gave the torque coefficient as

$$C_m = \frac{8\pi}{Re}\lambda, \tag{4.20}$$

where $\lambda = \lambda(\delta,b)$. He gave the value of λ for seven cases: for $\delta = 0.15$ with $b = 0.10$, 0.06 and 0.02, $\lambda = 1.30$, 2.30 and 4.78; for $\delta = 0.25$ with $b = 0.10$, 0.06 and 0.02, $\lambda = 1.22$, 2.08 and 4.27; for $\delta = 2.0$ with $b = 0.10$, $\lambda = 1.18$.

To compare the computational method with Schmieden's results,

calculations were done for the above combinations of aspect ratios and gaps for Reynolds numbers from 1 to 1000. For $\delta = 0.15$, a 201x31 grid was used; for $\delta = 0.25$, a 201x51 grid; and for $\delta = 2.0$, a 61x121 grid. All grids were uniform, and the symmetry boundary conditions were used.

Figures 4.44, 4.45 and 4.46 present the results for $\delta = 0.15$. For all three cases, the computational solutions run parallel to the analytical ones up to about Re = 500 at which point they begin to diverge. This is expected because the analytical result is good only for low Reynolds number; it is remarkable, though, that the solutions don't diverge until Re = 500. It can also be seen that as the gap decreases the difference between the solutions increases: at Re = 1 and b= 0.10, 0.06 and 0.02, the differences are about 10%, 30% and 100%, respectively.

Figures 4.47, 4.48 and 4.49 present the results for $\delta = 0.25$. Again, the solutions are parallel until, in these cases, Re = 100 or 500. At Re = 1 and b = 0.10, 0.06 and 0.02, the differences between the solutions are about 10%, 50% and 100%, respectively.

The above results show that the difference between the analytical and computational solutions increases as b decreases. This is because the creeping flow approximation breaks down as b decreases. At the tip of the rotating disk $v = R_d$, while at the sidewall, a distance b away, $v = 0$; therefore, $\partial v/\partial r = O(R_d/b)$ in the region near the gap. As b decreases, $\partial v/\partial r$ increases, and the velocity gradients are no longer small, i.e., the boundary conditions in

the gap violate the assumption of creeping flow.

Figure 4.50 presents the results for $\delta = 2.0$ and $b = 0.10$. Again, the solutions are parallel until $Re = 100$. At $Re = 1$, the difference between the computational and analytical solutions is 35%; for $\delta = 0.15$ and 0.25, the corresponding differences (for $b = 0.10$) were 10% in both cases. Therefore, Schmieden's solution also breaks down as the aspect ratio increases.

The reason for this is apparent from Figures 4.51, 4.52 and 4.53 which give the velocity profiles of v for $\delta = 0.15$, 0.25 and 2.0 with $Re = 1$ and $b = 0.10$. These were obtained from the computational work. For $\delta = 2.00$, a boundary layer has formed on the rotating disk with high velocity gradients at $z = 1$; therefore, the assumption of creeping flow is not valid. For $\delta = 0.10$ and 0.25, the velocity gradients are moderate because the proximity of the endwall prevents a boundary layer from forming; therefore, the creeping flow assumption is valid.

To summarize, the analytical solution of Schmieden fails in three ways: as Reynolds number increases, the assumption of creeping flow breaks down; as the gap decreases, the boundary conditions in the gap violate the assumption of small velocity gradients; as the aspect ratio increases, a boundary layer forms on the rotating disk, and the velocity gradients in the boundary layer violate the assumption of small gradients.

4.4 Torque Results

This section contains a discussion of the torque coefficient results obtained by the computational method described in Chapter 3. The core of this work is a set of ten graphs of C_m versus Re. Each graph is for a different aspect ratio and consists of a family of curves for different gaps. In most cases, the grids used were uniform (exceptions will be noted where appropriate) and were as high as 201x101. The grid size was chosen so as to obtain a square uniform grid as the aspect ratio changed, and to place a node point on the edge of the disk. In addition, it was found by numerical experiments that to obtain consistent results, it was necessary to place grid points inside the gap and to use a finer grid for high Reynolds number. The Couette flow boundary conditions were used for all calculations. All the data are given in Appendix D.

In Figure 4.54, the torque curves for $\delta = 0.02$ with b = 0.10, 0.05, 0.025, and 0.001 are given. For each b, C_m was determined for Re = 1, 10, 100, 500, 1000, 5000 and 10000. All grids were uniform except for b = 0.001, where the stretching transformation, given by equations (3.15), was applied to a 801x17 grid. Without this method, it would have been necessary to use an even finer grid for such a low value of b.

The torque curves in Figure 4.54 are straight, decrease as the Reynolds number increases and increase as the gap decreases; therefore, the torque coefficient is a function of the Reynolds number and the gap.

The relationship of C_m and b, however, is complicated by the changing

area of the disk. In equation (3.34), the upper limit of the integral is R_d, and as b decreases, R_d increases; however, Re, since it is based on R_c, remains constant. Therefore, C_m increases because the disk size is increasing. Even when this is taken into account, however, there is still an increase in C_m. For example, at Re = 1 and b = 0.10, C_m = 206.8, and at b = 0.001, C_m = 339.6. Dividing C_m by the area of the disks, and taking the ratios of these values gives a value of 1.33. As a result, even when the difference in area is taken into account, the torque coefficient at b = 0.001 is 33% greater than at b = 0.10. This is the edge effect caused by the influence of the sidewall.

The reason for the edge effect is apparent from the following: at the edge of the disk, $v = R_d$, but at the sidewall, v = 0; therefore, as the gap decreases, the velocity gradient--and with it the torque--increases.

Schmieden [S2] showed this trend clearly, although his values were wrong for small gaps. Pao [P2], citing Schmieden, was aware that C_m increases as b decreases. He also stated that the edge effect is less important for increasing Reynolds number; however, what he based this on is not clear, since Schmieden's results are only valid for low Reynolds number. From Figure 4.54, it is clear that the edge effect is important for all Reynolds numbers.

It can now be seen that the results shown in Figure 4.30 are caused by implicitly varying the gap. In equation (4.3), r_0 is taken as the second last node point. For 21x21, 41x41 and 81x81 grids this gives r_0 = 0.05, 0.025 and 0.0125, respectively. As r_0 decreases, $\partial v/\partial z$ at the second last node increases, which

is the same effect caused by decreasing the gap. Pao's method is grid dependent because it implicitly varies the gap. Of course, it does not treat the gap in a systematic way; therefore, while the trend is correct, the values are wrong. For example, for Re = 1 and b = 0.05, Pao's method gives $C_m = 45.39$, while the method developed here gives $C_m = 29.73$. At b = 0.025 the respective values are 56.32 and 40.21.

The torque coefficient is plotted as a function of b in Figure 4.55 for various Reynolds numbers. These curves more clearly illustrate the effect of the Reynolds number and the gap on C_m: C_m decreases as the Reynolds number increases and increases as the gap decreases. The problem with Figure 4.55 is that it is on semi-log scale, and details of the effect of b on C_m are difficult to see. Figure 4.56 gives the curve for Re = 1 of Figure 4.55 on a normal scale. The edge effect is now clear: C_m increases as b decreases, and for small b the increase in C_m accelerates.

This effect was reported by Lugt and Haussling [L4]. Using a stretching transformation on a 41x41 grid, they moved the second last grid point on the disk closer to the wall. They showed a significant increase in the term $(1/r)\partial v/\partial z$, thereby showing the effect of the singularity of equation (4.2). Again, the general trend they showed was correct, but to get proper values, the boundary conditions in the gap must be dealt with carefully.

Next a set of torque curves are given for δ = 0.10, 0.25 and 0.50 in Figures 4.57, 4.58, 4.59. They give the torque coefficient for b = 0.10, 0.05,

0.025 and 0.01. The graphs can be divided into three regions: two straight lines joined by a curved crossover region. The crossover region is around Re = 750, 100 and 100, respectively, for the three figures. This corresponds well with the experimental results of Daily and Nece [D1] shown in Figures 4.38 and 4.39.

Figure 4.54, for $\delta = 0.02$, shows no crossover region. From Figure 4.38, however, it can be seen that there is one around Re = 20 000.

The figures also show that the torque coefficient varies with the gap. For the first region, the curves are parallel and well spaced; for example, at $\delta = 0.50$ and Re = 1, the difference between C_m at b = 0.10 and 0.01 is about 175%. For region three, however, the difference between C_m, while still significant, has decreased; for example, at $\delta = 0.50$ and $Re = 10^4$, the difference between C_m at b = 0.10 and 0.01 has dropped to about 35%. Recall, Pao [P2] concluded that the end effect decreases as the Reynolds number increases. This is true, but it is still significant.

The next set of torque curves, for $\delta = 0.75$ and 1.00, are given in Figures 4.60, and 4.61. For these, and for all subsequent graphs, b = 0.10, 0.05 and 0.025. The same general trends seen above are apparent here. The start of the crossover region has stabilized to about Re = 100; this will be true for all remaining torque graphs.

The next graph, Figure 4.62, is for $\delta = 1.50$. In this case C_m was only calculated up to Re = 5000. An attempt was made at Re = 10 000, but it failed,

even though the grid was refined to 161x241.

In the experimental work of Escudier [E1], the region of stable solutions with respect to aspect ratio and Reynolds number was mapped out. Above about $\delta = 1.75$, the flow becomes unstable for Reynolds number above 2500; specifically, the separation bubble on the axis of symmetry oscillates. For an aspect ratio of 1.50, Escudier gives no indication of stability problems; however, the maximum Reynolds number of his experiments was about 3500. It is, therefore, not clear why the computational method did not converge at Re = 10 000.

The final set of torque coefficient curves are given in Figures 4.63, 4.64 and 4.65, for $\delta = 2.0$, 2.5 and 3.0, respectively. For these values of aspect ratios, the flow is within the area that Escudier identified as oscillating; therefore, these curves were only calculated for Reynolds numbers up to 2500. The same general trends can be seen as in the above cases. This completes the overall description of the torque curves. Next, some specific points will be made.

First, the range of validity with respect to aspect ratio of equations (4.18) and (4.19) of Daily and Nece [D1] will be considered. In Figure 4.66, a comparison is made with equation (4.18) and the computational results for Re = 10, b = 0.225 and $\delta = 0.02$, 0.10, 0.25 and 0.50. These values should place C_m in Regime I. The two curves quickly diverge, and at $\delta = 0.10$ the difference is already about 20%. It is concluded that the linear velocity profile assumption is only valid for very small aspect ratios. It should be noted,

though, that Daily and Nece never reported experimental results for Reynolds numbers this low.

In Figure 4.67, the same comparison is made for equation (4.19) for $Re = 10^4$ and $b = 0.025$. The aspect ratios start at 0.10, since this is the first value for which Regime II should hold, and goes up to 3.00. Agreement is extremely good for all values. The maximum deviation occurs at $\delta = 3.0$ and is about 10%. Considering the fact that they only did experiments for $\delta = 0.0127$ to 0.0217, this agreement is remarkable. One point of caution should be made. For their experiments, they used $b = 0.006\,33$, and as seen above, the effect of decreasing b is very significant; therefore, it would be questionable if equation (4.19) would still hold for very low values of b.

The effect of the aspect ratio on C_m is considered next. In Figure 4.68, C_m is plotted as a function of δ for $Re = 100$, 500 and 1000 and $b = 0.025$. After an initial drop, C_m becomes constant. This does not mean, however, that the actual torque is constant. From equation (3.34) it is seen that the torque coefficient is inversely proportional to both δ and Re. In Figure 4.69, the function $C_m\delta Re$, the torque integral, is plotted as a function of δ for $Re = 500$ and $b = 0.025$. It is monotonically increasing but increases most rapidly at the beginning and then levels off.

What is happening is that at the lowest aspect ratios, the profile is approximately linear and therefore $\partial v/\partial z$ at the disk is a minimum. As the aspect ratio increases, the velocity profile becomes nonlinear resulting in a rapid

increase in the torque requirement. At higher aspect ratios, though, a boundary layer forms on the disk. It might be expected that above a certain value of δ, the boundary layer would remain constant because it is no longer influenced by the endwall, and, therefore, the torque integral should be constant. It must be considered, however, that the rotating disk is what drives the flow: fluid from the rotating disk is pushed out to the sidewall, down the sidewall, along the endwall and up to the rotating disk again. As the aspect ratio gets larger, it is necessary to drive the fluid a further distance--therefore, the torque integral increases with the aspect ratio.

Finally, a comparison with the Schultz-Grunow correlation is made. Schultz-Grunow [S3] derived a formula for the torque coefficient by assuming that there are boundary layers on the disk and endwall and that the remaining core of the fluid rotates at a constant angular velocity. He ignored the effect of the sidewall. From these assumptions, he obtained the correlation

$$C_m = \frac{2.67}{\sqrt{Re}}(1-b)^4. \qquad (4.21)$$

This formula is of considerable importance because it is often used.

Figures 4.70, 4.71 and 4.72 are a comparison of the Schultz-Grunow correlation with the computational results for δ = 0.10, 0.50 and 3.0 with b = 0.025. The two curves do not get close until after the crossover region is passed; then they run parallel. At Re = 1000, the differences between the two curves are about 5%, 20% and 25% for δ = 0.10, 0.50 and 3.0, respectively.

Therefore, for Re ≥ 1000, equation (4.21) gives very good results for small aspect ratios, and for $\delta > 0.10$, it gives an acceptable first approximation. At lower Reynolds numbers, the correlation is invalid. The correlation would not be applicable for very small aspect ratios because there would be no separated boundary layers.

4.5 Primary and Secondary Flow Rates

This section will deal with the primary and secondary volumetric flow rates, Q_p and Q_s, defined by equations (3.29) and (3.32). These values, which give measures of the rate of fluid flow, are important for the design of mixing vessels and in understanding fluid flows in the environment. In addition, though heat transfer was not considered in this work, they can give a qualitative indication of heat transfer behaviour. For example, knowledge of where the flow rates are maximum and minimum can give an indication of where the heat transfer would be maximum or minimum. The ranges of Reynolds number, aspect ratio and gap are the same as in Section 4.4, and, therefore, this information is not repeated.

For the calculation of Q_s, equation (3.33) provided a useful check on equation (3.32). The agreement obtained between the two was very good. In most cases the difference was well below 1% but sometimes was as high as 8%; for example, at Re = 1 and $\delta = 0.02$, the difference was 8%, while at $Re = 10^4$ and $\delta = 3.0$, it was 0.03%. It should also be noted that, in addition to

the expected numerical errors, there is an intrinsic error in the calculation of Q_s. Both equations depend on the vortex centre, where ψ is maximum, but unless the vortex centre is exactly on a node point, its exact value and location cannot be determined.

First the effect of the gap on Q_p and Q_s is considered. In Figure 4.73 and 4.74, Q_p and Q_s are given for $\delta = 0.25$ with b = 0.10, 0.05, 0.025 and 0.01. As can be seen, Q_p and Q_s show some variation with b: at Re = 1.0, the variation is about 5% for both Q_p and Q_s. At this point, C_m had a variation of 60%. Therefore, Q_p and Q_s are weakly dependent on b. This is true for all aspect ratios, and, therefore, these graphs are not repeated for the other values of δ.

There is one exception to this. At $\delta = 0.02$ the variation is about 15% for Q_s at high Reynolds numbers and up to 90% at lower values; however, at lower Reynolds numbers, Q_s is of the order 10^{-8}, and, therefore, this large variation is probably not significant. The complete set of data is available in Appendix D.

In Figures 4.75 to 4.84, Q_p and Q_s are given for b = 0.025. Each figure is for a different aspect ratio from 0.02 to 3.0. First consider the variation of Q_p with respect to Reynolds number. This variation is small for lower aspect ratios, while at higher values it is moderate. For example, at $\delta = 0.25$, Figure 4.77, the difference between minimum and maximum values of Q_p is about 20%, but at $\delta = 3.0$, Figure 4.84, this difference is about 45%.

The variation of Q_s is much larger. At $\delta = 0.25$ and 3.0, the difference

between minimum and maximum is about 100% in both cases. In addition, for δ = 0.10, 0.025 and 0.50, Q_s has a sharp maximum, while for higher aspect ratios, this is replaced by a plateau. For δ = 0.02, Q_s is monotonically increasing. The primary flow rate shows a distinct minimum, and this minimum is associated with the maximum of Q_s and the crossover region of C_m described in Section 4.4. In addition, Q_p is always greater than Q_s except for δ = 0.10, 0.25 and 0.50 where Q_s exceeds Q_p but only at the maximum of Q_s.

The graphs just considered are fairly coarse: the Reynolds number varies from 1 to 10^4 with only seven data points. In Section 4.3.4, results were presented for δ = 0.02 and 0.10, with b = 0.0063 using the symmetry boundary condition where it was necessary to calculate more points. The results of Q_p and Q_s for these cases are presented in Figures 4.85 and 4.86.

For δ = 0.02, Figure 4.85 (cf. Figure 4.75), Q_s continues to increase until Re = 75 000, at which point a slight drop occurs. Q_p is monotonically decreasing. For δ = 0.10, Figure 4.86 has the same characteristic as shown in Figure 4.76: maximum Q_s still occurs at Re = 5000 but is not as sharp. At Re = 70 000, Q_s has decreased to about the same value it had at Re = 500.

Finally, in Figures 4.87 and 4.88, the effect of aspect ratio on Q_p and Q_s is shown. The results are for b = 0.025 and Re = 100, 500 and 1000. Figure 4.87 shows that Q_p increases at lower aspect ratios and then reaches a constant value. All the curves meet around δ = 0.75, and above this point the higher the Reynolds number the higher Q_p. Figure 4.88 shows the effect on Q_s.

This time the rate of increase at small aspect ratios is much greater, and a constant value is achieved faster. For high aspect ratios, the lowest value of Q_s is at Re = 1000, followed by 100 and then 500. This is because Q_s is maximum at Re = 500.

CHAPTER 5: CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions for the Optimization Study

1. The optimization study indicated that there are four regions in the ω-Δt space: a converging region, an overflow region, an underflow region and a nonconverging region. The overflow region is a result of numerical instability, while the underflow region is a result of the fluid-at-rest initial conditions.

2. The optimum iteration parameters, Δt_{opt} and ω_{opt}, depended on the grid spacing and the Reynolds number; they tended to occur near the overflow region. As the Reynolds number increased, Δt_{opt} increased, and as the grid spacing decreased, Δt_{opt} increased. The value of ω_{opt} depended on the Reynolds number and the grid spacing in a more complex fashion.

3. The rate of convergence is very sensitive to iteration parameter variation near the optimum point.

4. The number of iterations required for convergence is dependent on ω and Δt. It is not possible to determine the convergence behaviour of a computational fluid problem by considering the finite-difference equations separately.

5. A technique was developed to help find the optimum iteration parameter.

5.2 Recommendations for the Optimization Study

1. A similar optimization study for an ADI-ADI solver would be useful.

2. An investigation of what is happening in the nonconverging region should be done, i.e., is it oscillating indefinitely, converging very slowly or diverging very slowly?

3. A study of the use of nonzero initial conditions, especially for high Reynolds numbers and high grid densities, should be done. The effect on Δt_{opt} and ω_{opt} would be important.

5.3 Conclusions for the Rotating Flow Problem

1. Previous computational determinations of the torque coefficient were inaccurate, because the singularity of the velocity gradient where the rotating disk meets the sidewall was not dealt with adequately. A method was developed to overcome this problem, and this method was shown to be accurate based on the experimental results of Daily and Nece [D1]. This method was simple and easy to apply.

2. The most important feature in dealing with the torque calculation is to treat the boundary conditions in a consistent fashion.

3. The computational results were compared with the analytical work of Schmieden [S2] who assumed creeping flow. Schmieden's results were found to break down as the Reynolds number increased, the gap decreased and the aspect ratio increased.

4. The torque coefficient was a function of the Reynolds number, the aspect ratio and the gap. For high aspect ratios, however, it tended to a constant.

5. Extrapolating the empirical formula of Daily and Nece for Regime II, equation (4.19), beyond the aspect ratios of their experimental work gave very good results for high aspect ratios. This was not the case for Regime I, equation (4.18), which was only valid for $\delta < 1$.

6. The commonly used Schultz-Grunow correlation was valid only for $Re \geq 1000$. The error varied from 5% to 25%.

7. The primary and secondary volumetric flow rates, Q_p and Q_s, are functions of the Reynolds number and the aspect ratio. For higher aspect ratios, they tended to a constant. They are weak functions with respect to the gap. Q_p is greater than Q_s, except for a few points at low aspect ratios.

8. As the Reynolds number was varied, Q_p and Q_s varied in phase with each other: as one decreased, the other increased and vice versa.

5.4 Recommendations for the Rotating Flow Problem

1. To ensure that a grid point occurred at the edge of the rotating disk, it was necessary, in many cases, to use a high number of grid points. The development of a technique to allow a grid point to be forced onto the plate edge for an arbitrary gap would be useful. On the other

hand, an extrapolation technique might also work; however, it might be difficult to develop because $\partial v/\partial z$ tends to be very steep at the disk edge.

2. A more thorough investigation should be carried out of the flow fields and how they compare with Regime I and II of Daily and Nece.

3. The actual boundary conditions in the gap are not clear. A more through investigation of this would be useful. This could be done numerically, by having a step in the computational domain to represent the disk. It could also be done experimentally. Because of the nature of the flow, an LDA study would be the most useful.

REFERENCES

A1 Anderson, D.A., Tannehill, J.C. and Pletcher, R.H., Computational Fluid Mechanics and Heat Transfer, Hemisphere Publishing Corporation, Washington, 1984.

B1 Batchelor, G.K., "Note on a class of solutions of the Navier-Stokes equations representing steady rotationally-symmetric flow", Quart. J. Mech. Appl. Math., **4**:29-41, 1951.

B2 Benjamin, A.S., "Finite difference solution of recirculating flow in a two-dimensional cavity at high Reynolds number", Ph.D. Thesis, UCLA, 1977.

B3 Benjamin, A.S. and Denny, V.E., "On the convergence of numerical solutions for 2-D flows in a cavity at large Re", J. Computational Phys., **33**: 340-358, 1979.

B4 Bertelà, M. and Gori, F., "Laminar flow in a cylindrical container with a rotating cover," J. Fluids Engr., **104**:31-39, 1982.

B5 Brazier, P.H., "An optimum SOR procedure for the solution of elliptic partial differential equations with any domain or coefficient set," Computer Methods in Applied Mech. and Engr., **3**:335-347, 1974.

B6 Briley, W.R., "A numerical study of laminar separation bubbles using the Navier-Stokes equations", Report J110614-1, United Aircraft Research Laboratories, East Hartford, Connecticut, 1970.

B7 Briley, W.R., "A numerical study of laminar separation bubbles using the Navier-Stokes equation", J. Fluid Mech., **47**:713-736, 1971.

B8 Brown, G.L. and Lopez, J.M., "Axisymmetric vortex breakdown. Part 2. Physical mechanisms," J. Fluid Mech., **221**:553-576, 1990.

C1 Carré, B.A., "The determination of the optimum accelerating factor for successive over-relaxation," Computer J., **4**:73-78, 1961.

C2 Cochran, W.G., "The flow due to a rotating disc," Proc. Cambridge Phil. Soc., **30**:365-375, 1934.

D1 Daily, J.W. and Nece, R.E., "Chamber dimension effects on induced flow and frictional resistance of enclosed rotating disks," J. Basic Engr.,

82:217-232, 1960.

D2 Daniels, W.A., Johnson, B.V. and Graber, D.J., "Aerodynamic and torque characteristics of enclosed co/counterrotating disks," J. Turbomachinery, **113**:67-74, 1991.

D3 Dijkstra, D. and Heijst, G.J.F. van, "The flow between two finite rotating disks enclosed by a cylinder," J. Fluid Mech., **128**:123-154, 1983.

D4 Douglas, J., "On the numerical integration of $\partial^2 u/\partial x^2 + \partial^2 u/\partial y^2 = \partial u/\partial t$ by implicit methods," J. Soc. Indust. Appl. Math., **3**:42-65, 1955.

D5 Duck, P.W., "On the flow between two rotating shrouded discs," Computers & Fluids, **14**:183-196, 1986.

E1 Escudier, M.P., "Observations of the flow produced in a cylindrical container by a rotating endwall," Experiments in Fluids, **2**:189-196, 1984.

F1 Frankel, S.P., "Convergence rates of iterative treatments of partial differential equations," Math Tables and Other Aids to Computation, **4**:65-75, 1950.

G1 Ghia, U., Ghia, K.N., and Shin, C.T., "High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method," J. Computational Phys., **48**:387-411, 1982.

G2 Greenspan, H.P., The Theory of Rotating Fluids, Cambridge University Press, 1968.

G3 Gustafsson, B., "On difference approximations to hyperbolic differential equations over long time intervals," SIAM J. Numer. Anal., **6**:508-522, 1969.

H1 Hirt, C.W., "Heuristic stability theory for finite-difference equations," J. Computational Phys., **2**:339-355, 1968.

H2 Hudson, A.J. and Eibeck, P.A., "Torque measurements of corotating disks in an axisymmetric enclosure," J. Fluids Engr., **113**:648-653, 1991.

I1 IBM Corporation, VS FORTRAN Version 2. Language and Library Reference (Release 4), SC26-4221-04, 1989.

K1 Kármán, T. von, "Über laminare und turbulente Reibung", ZAMM, **1**:233-

252, 1921.

K2 Kármán, T. von, "On laminar and turbulent friction," NACA TM 1092, 1946.

L1 Lehmkuhl, G.D. and Hudson, J.L., "Flow and mass transfer near an enclosed rotating disk: experiment," Chemical Engr. Science, **26**:1601-1613, 1971.

L2 Lopez, J.M., "Axisymmetric vortex breakdown. Part 1. Confined swirling flow," J. Fluid Mech., **221**:533-552, 1990.

L3 Lugt, H.J. and Abboud, M., "Axisymmetric vortex breakdown with and without temperature effects in a container with a rotating lid," J. Fluid Mech., **179**:179-200, 1987.

L4 Lugt, H.J., and Haussling, H.J., "Transient Ekman and Stewartson layers in a rotating tank with a spinning cover," IUTAM Symposium on Unsteady Boundary Layers, Laval University, Quebec, 1366-1396, 1971.

L5 Lugt, H.J. and Haussling, H.J., "Development of flow circulation in a rotating tank," Acta Mechanica, **18**:255-272, 1973.

L6 Lugt, H.J. and Haussling, H.J., "Axisymmetric vortex breakdown in rotating fluid within a container," J. Applied Mech., **49**:921-923, 1982.

M1 Mallinson, G.D. and De Vahl Davis, G., "The method of the false transient: application to fluid mechanics," Report ARL/SM. 331, Department of Supply, Australian Defence Scientific Service, Aeronautical Research Laboratories, Melbourne, 1971.

M2 Mallinson, G.D. and De Vahl Davis, G., "The method of the false transient for the solution of coupled elliptic equations," J. Computational Phys., **12**:435-461, 1973.

M3 Mellor, G.L., Chapple, P.J. and Stokes, V.K., "On the flow between a rotating and a stationary disk," J. Fluid Mech., **31**:95-112, 1968.

N1 Nallasamy, M. and Prasad, K.K., "On cavity flow at high Reynolds numbers," J. Fluid Mech., **79**:391-414,1977.

O1 O'Brien, G.G., Hyman, M.A. and Kaplan, S., "A study of the numerical solution of partial differential equations," J. Mathematics and Physics,

29:223-251, 1950.

P1 Pao, H.-P., "A numerical computation of a confined rotating flow," J. Applied Mech., **37**:480-487, 1970.

P2 Pao, H.-P., "Numerical solution of the Navier-Stokes equations for flows in the disk-cylinder system," Phys. Fluids, **15**:4-11, 1972.

P3 Peaceman, D.W. and Rachford, H.H., "The numerical solution of parabolic and elliptic differential equations," J. Soc. Indust. Appl. Math., **3**:28-41, 1955.

P4 Pearson, C.E., "Numerical solutions of the time-dependent viscous flow between two rotating coaxial disks," J. Fluid Mech., **21**:623-633, 1965.

P5 Piesche, M., "Investigation of the flow in the impeller-side space of rotary pumps with superimposed throughflow for the determination of axial force and frictional torque," Acta Mechanica, **78**:175-189, 1989.

R1 Rasmussen, H., "Numerical solutions for steady viscous linearized flow between two finite rotating disks," ZAMP, **21**: 611-618, 1970.

R2 Roache, P.J., Computational Fluid Dynamics, Hermosa Publishers, Albuquerque, 1982.

R3 Rogers, M.H. and Lance, G.N., "The boundary layer on a disc of finite radius in a rotating fluid," Quart. J. Mech. and Applied Math., **17**:319-330, 1964.

R4 Ronnenberg, B., "Ein selbstjustierendes 3-Komponenten-Laserdoppleranemometer nach dem Vergleichsstrahlverfahren, angewandt für Untersuchungen in einer stationären zylindersymmetrischen Drehströmung mit einem Rückstromgebiet," Max-Planck-Institut für Strömungsforschung, Göttingen, Bericht 20/1977, 1977.

S1 Schlichting, H., Boundary-Layer Theory, Seventh Edition, McGraw-Hill Book Co., New York, 1979.

S2 Schmieden, C., "Über den Widerstand einer in einer Flüssigkeit rotierenden Scheibe," ZAMM, **8**:460-479, 1928.

S3 Schultz-Grunow, F., "Der Reibungswiderstand rotierender Scheiben in

Gehäusen," ZAMM, **15**:191-204, 1935.

S4 Stewartson, K., "On the flow between two rotating coaxial disks," Proc. Cambridge Phil. Soc., **49**:333-341, 1953.

T1 Theodorsen, T. and Regier, A., "Experiments on drag of revolving disks, cylinders, and streamline rods at high speeds," NACA Report No. 793, 1944.

T2 Tomlan, P.F. and Hudson, J.L., "Flow near an enclosed rotating disk: analysis," Chemical Engr. Science, **26**:1591-1600, 1971.

T3 Trapp, J.A. and Ramshaw, J.D., "A simple heuristic method for analyzing the effect of boundary conditions on numerical stability," J. Computational Phys., **20**:238-242, 1976.

V1 Vogel, H.U., "Experimentelle Ergebnisse über die laminare Strömung in einem zylindrischen Gehäuse mit darin rotierender Scheibe," Max-Planck-Institut für Strömungsforschung, Göttingen, Bericht 6/1968, 1968.

W1 Wachspress, E.L., "Optimum alternating-direction-implicit iteration parameters for a model problem," J. Soc. Indust. Appl. Math., **10**:339-350, 1962.

W2 Wachspress, E.L. and Habetler, G.J., "An alternating-direction-implicit iteration technique," J. Soc. Indust. Appl. Math., **8**:403-424, 1960.

W3 Warming, R.F. and Beam, R.M., "An extension of A-stability to alternating direction implicit methods," BIT, **19**:395-417, 1979.

W4 White, F.M., Fluid Mechanics, Second Edition, McGraw-Hill Book Co., New York, 1986.

Y1 Young, D., "Iterative methods for solving partial difference equations of elliptic type," Transactions of the American Mathematical Society, **76**:92-111, 1954.

Z1 Zandbergen, P.J. and Dijkstra, D., "Von Kármán swirling flows," Ann. Rev. Fluid Mech., **19**:465-491, 1987.2

TABLES

Figure	Ψ_{min}	Ψ_{max}	N_{min}	N_{max}
4.1	-1.154×10^{-1}	1.683×10^{-3}	20	20
4.5	-1.106×10^{-1}	2.636×10^{-3}	20	20
4.8	-3.207×10^{-3}	7.669×10^{-6}	20	20
4.10	-2.247×10^{-3}	6.879×10^{-6}	20	20
4.35	-1.369×10^{-2}	3.233×10^{-6}	20	10
4.36	-1.502×10^{-2}	1.679×10^{-6}	20	4
4.37	-3.920×10^{-3}	1.245×10^{-7}	20	20

Table 4.1 Data for Contour Plots (see equations (4.1))

	Re = 1000		Re = 3200	
	Ghia, et al. [G1]	Present	Ghia, et al. [G1]	Present
TLH			0.0859	0.0930
TLV			0.2057	0.2186
BLH	0.2188	0.2244	0.2844	0.3010
BLV	0.1680	0.1690	0.2305	0.2412
BRH	0.3034	0.3047	0.3406	0.3459
BRV	0.3536	0.3612	0.4102	0.4118
$\psi_{v.c.}$	-0.117929	-0.115354	-0.120377	-0.110633
$\zeta_{v.c.}$	2.04968	2.01604	1.98860	1.80959
$x_{v.c.}$	0.5313	0.53125	0.5165	0.515625
$y_{v.c.}$	0.5625	0.5625	0.5469	0.5390625

Table 4.2 Comparison of the Dimensions of Recirculation Regions Between Ghia, et al. [G1] and the Present Study (see Figures 4.1 and 4.5)

	Escudier [E1]	Lugt and Abboud [L3]	Lopez [L2]	Present
s_2/H	0.17	0.08	0.12	0.13
h_2/H	0.43	0.54	0.53	0.49
d_2/R_c		0.12	0.08	0.10
s_1/H	0.08	0.06	0.08	0.08
h_1/H	0.24	0.24	0.24	0.22
d_1/R_c		0.16	0.13	0.15

Table 4.3 Comparison of the Dimensions of the Recirculation Regions Between Experimental and Computational Results and the Present Study for Re = 1994 and δ = 2.5 (see Figure 4.8)

	Escudier [E1]	Lugt and Abboud [L3]	Present
s_2/H	0.25	0.30	0.32
h_2/H	0.43	0.32	0.36
d_2/R_c		0.11	0.12
s_1/H	0.04	0.07	0.05
h_1/H	0.18	0.15	0.16
d_1/R_c		0.19	0.17

Table 4.4 Comparison of the Dimensions of the Recirculation Regions Between Experimental and Computational Results and the Present Study for Re = 2752 and δ = 3.25 (see Figure 4.10)

Figure	Re	Grid	ω (Δω)	Δt (Δ[Δt])	t_{total}(h)
4.12	10	21x21	0.50-2.00 (.01)	.000-.100 (.0025) .105-.300 (.005) .310-1.00 (.010)	26
4.13	10	21x21	1.00-2.00 (.01)	.000-.125 (.001)	36
4.14	100	36x36	1.00-2.00 (.10)	.000-.350 (.005)	30
4.15	100	51x51	1.00-2.00 (.10)	.000-.080 (.005) .090-.150 (.005)	25
4.16	100	66x66	1.00-2.00 (.10)	.000-.075 (.005)	62
4.18	100	81x81	1.00-2.00 (.10)	.000-.050 (.005)	54
4.19	100	101x101	1.00-2.00 (.10)	.000-.030 (.005)	105
4.20	250	51x51	1.00-2.00 (.10)	.000-.200 (.005) .210-.400 (.010)	53
4.21	250	81x81	1.00-2.00 (.10)	.000-.070 (.005) .080-.150 (.010)	95
4.22	500	81x81	1.00-2.00 (.10)	.000-.120 (.005) .130-.250 (.010)	137
4.23	750	81x81	1.00-2.00 (.10)	.000-.200 (.005) .210-.400 (.010)	223

Table 4.5 Mesh and CPU Data for the Iteration Contour Graphs

Figure	Re	Grid	Δt_{opt}	ω_{opt}	I_{opt}	t_{opt}(min)
4.12	10	21x21	0.019	1.72	66	0.025
4.14	100	36x36	0.077	1.74	146	0.17
4.15	100	51x51	0.042	1.77	246	0.59
4.16	100	66x66	0.027	1.79	359	1.49
4.18	100	81x81	0.018	1.80	488	3.10
4.19	100	101x101	0.012	1.81	682	6.88
4.20	250	51x51	0.090	1.78	394	0.94
4.21	250	81x81	0.035	1.84	807	5.04
4.22	500	81x81	0.048	1.95	886	5.53
4.23	750	81x81	0.064	1.95	1096	6.83

Table 4.6 Data for the Optimum Point for the Iteration Contour Graphs

FIGURES

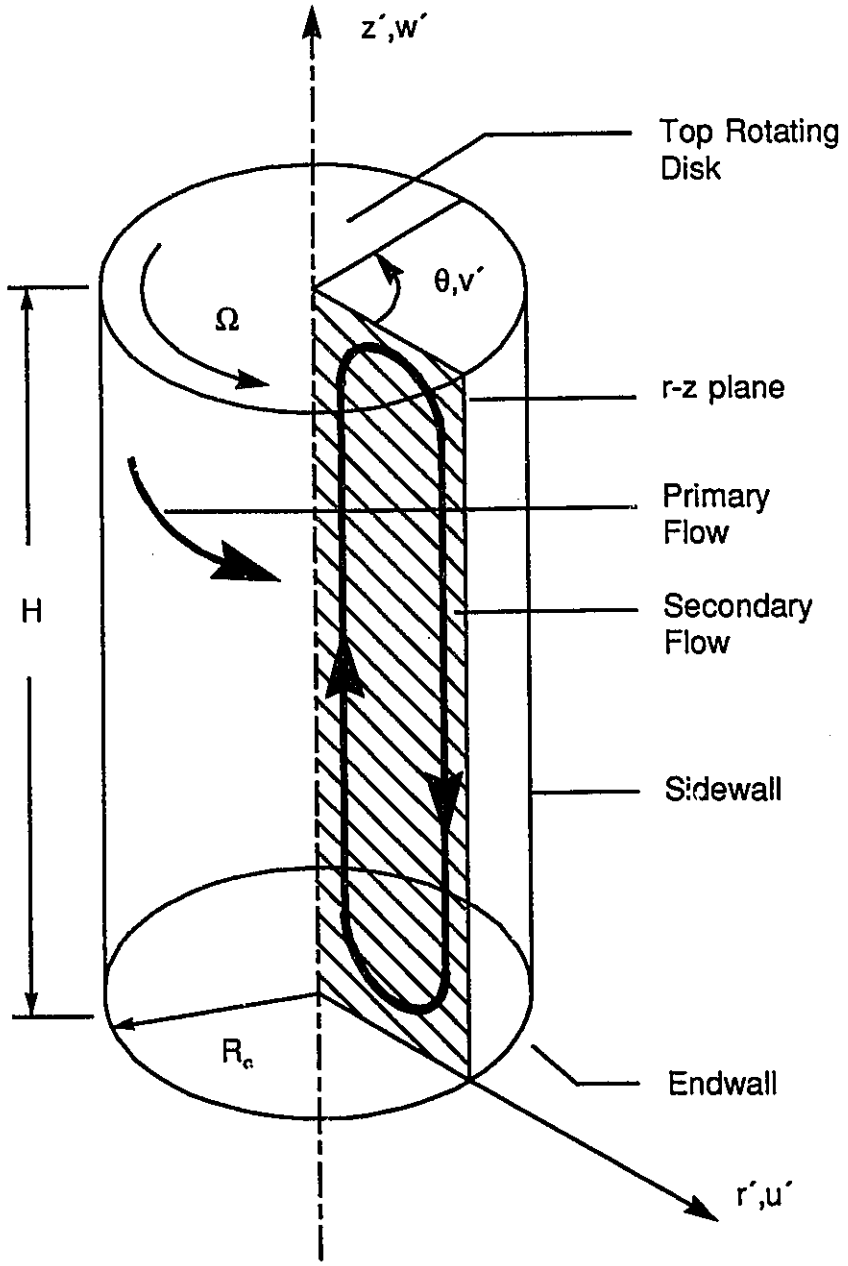

Figure 1.1 Geometry of the Rotating Flow Problem

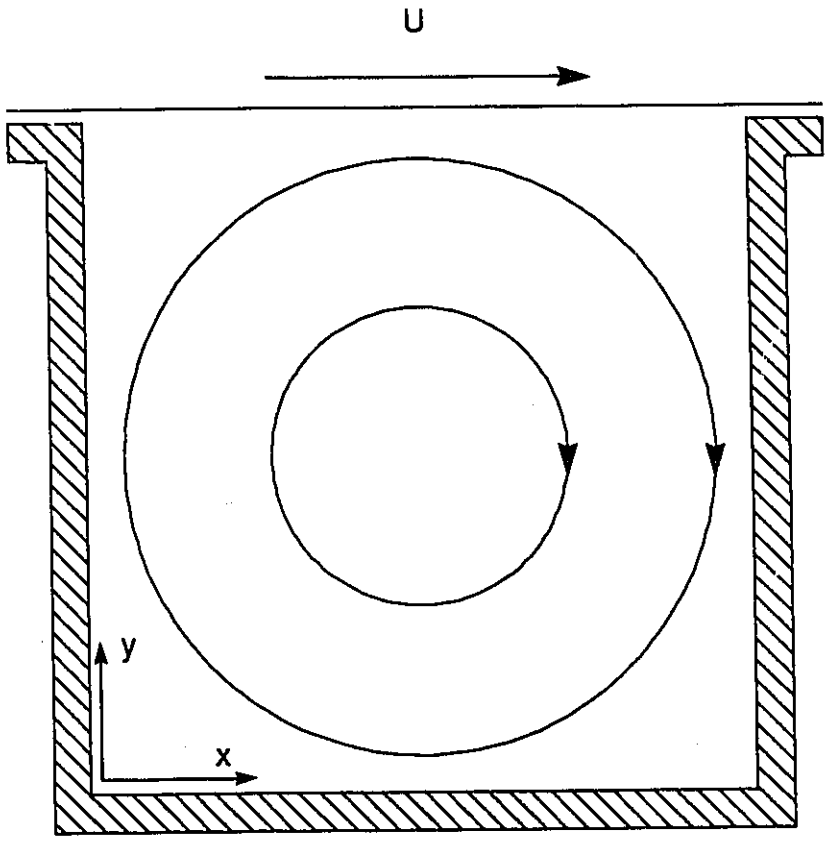

Figure 1.2 Geometry of the Cavity Flow Problem

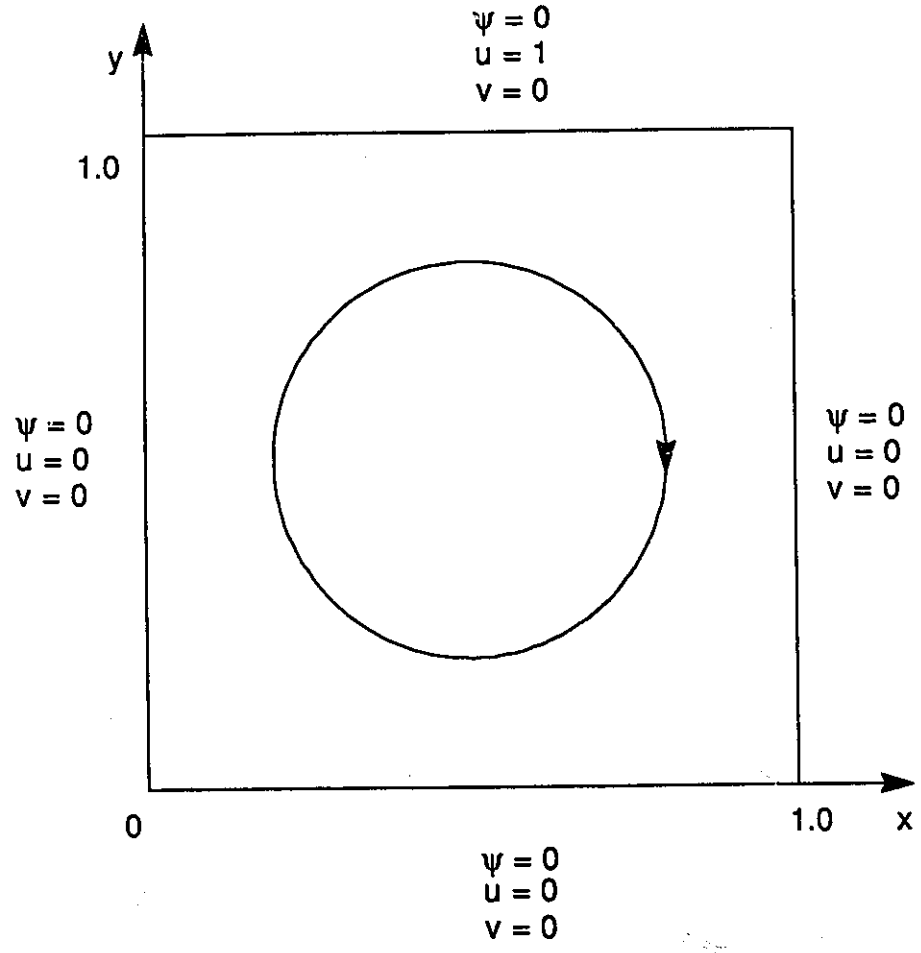

Figure 3.1 Computational Domain of the Cavity Problem

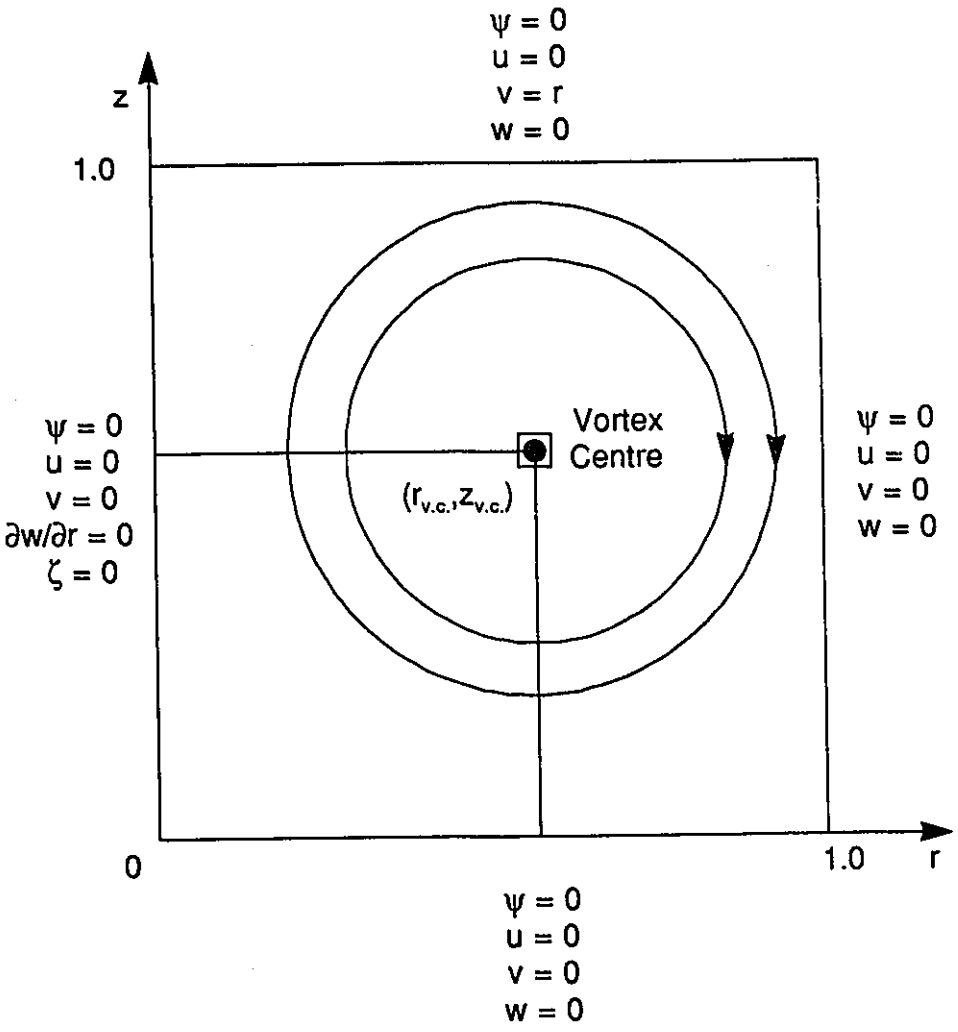

Figure 3.2 Computational Domain of the Rotating Flow Problem

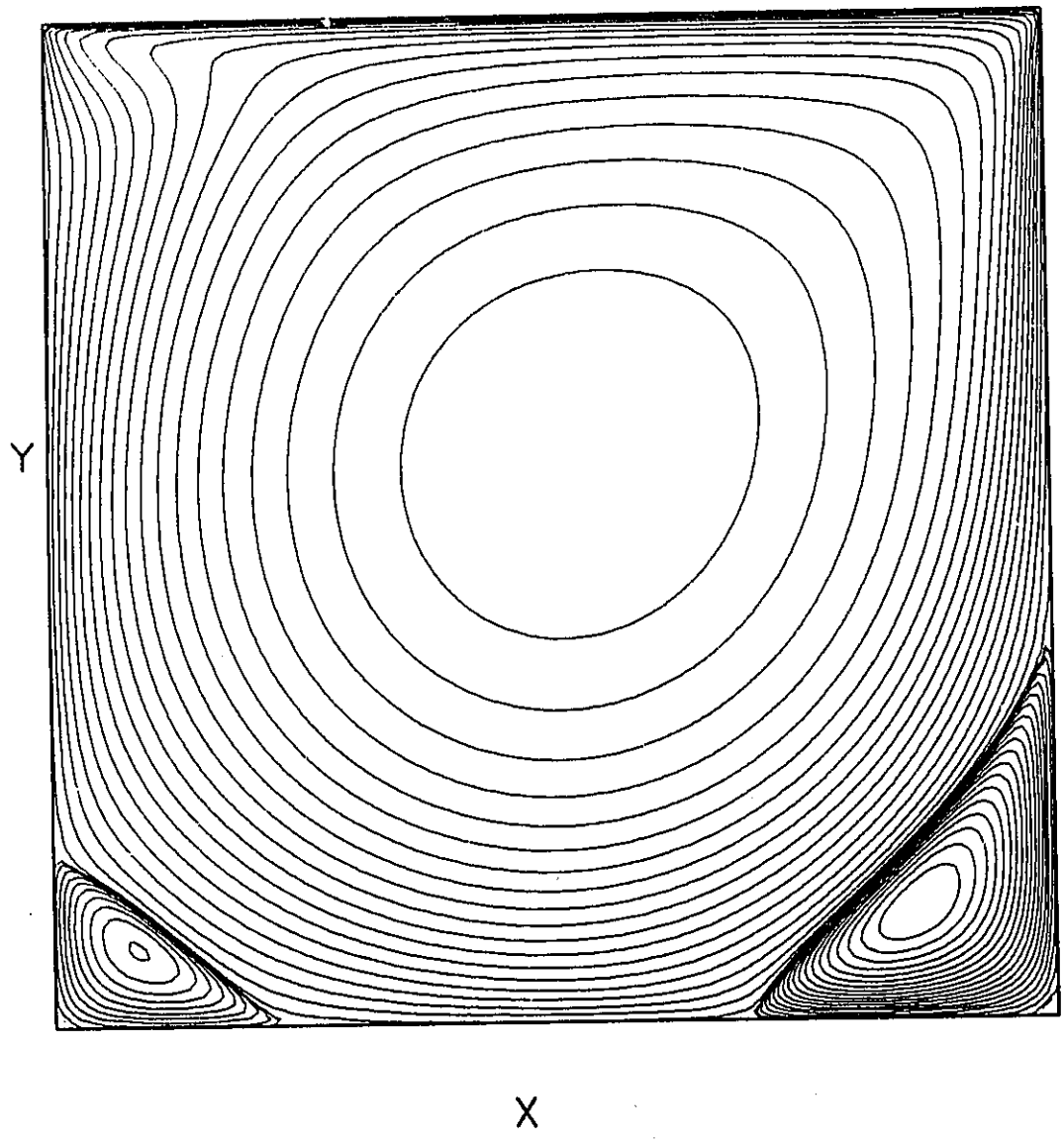

Figure 4.1 Cavity Flow Stream Function Contours for Re = 1000 on a 129x129 Uniform Grid

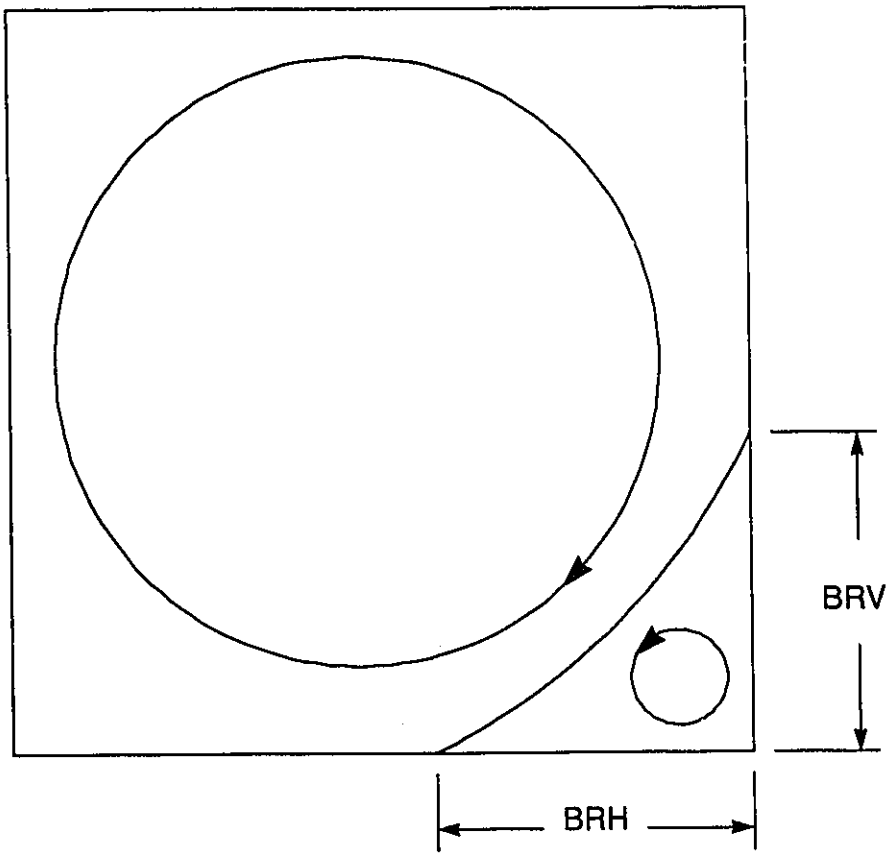

Figure 4.2 Nomenclature for the Dimension of the Cavity Flow Recirculation Regions

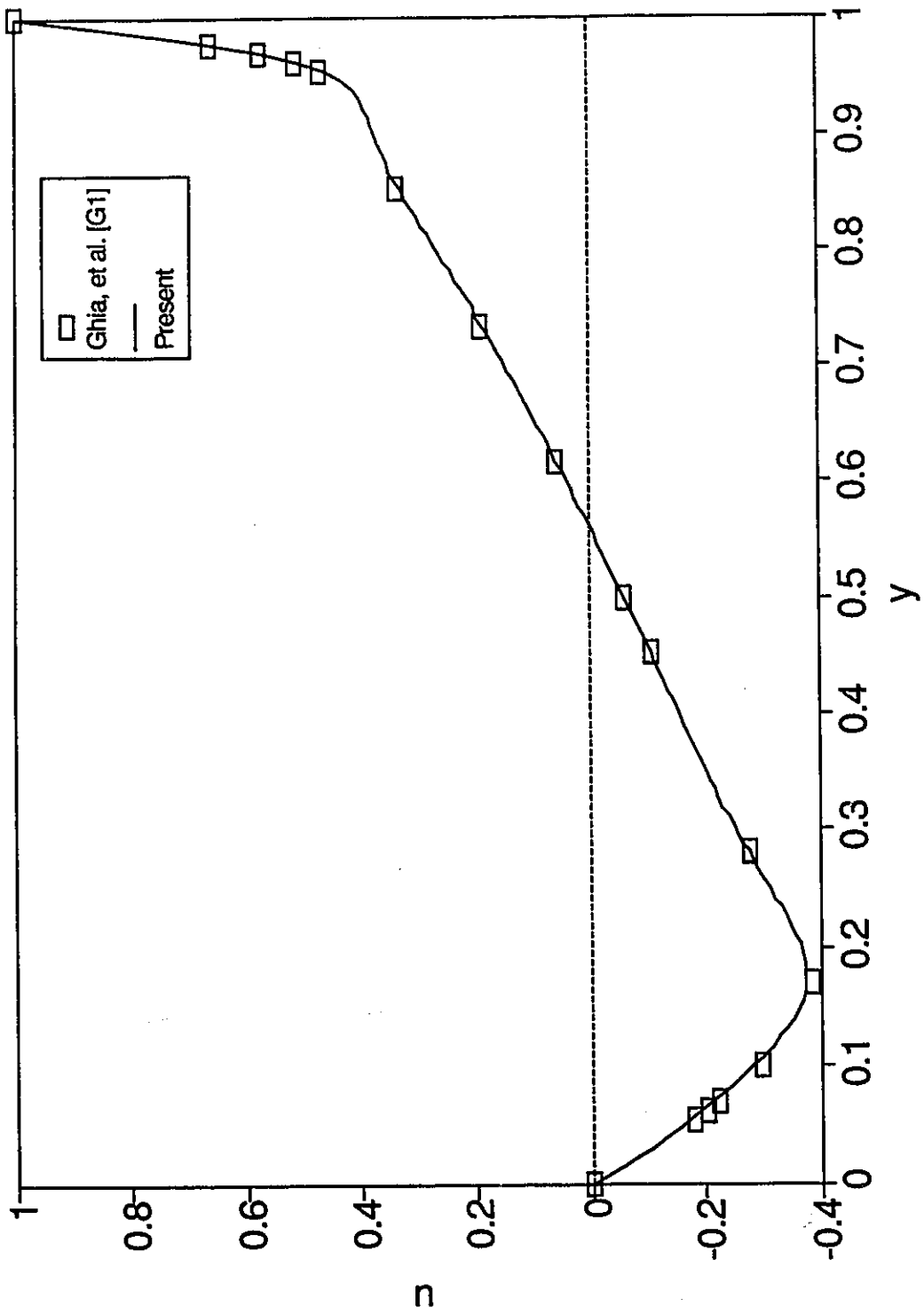

Figure 4.3 u-component of Velocity Along the Line x = 0.5 for Re = 1000

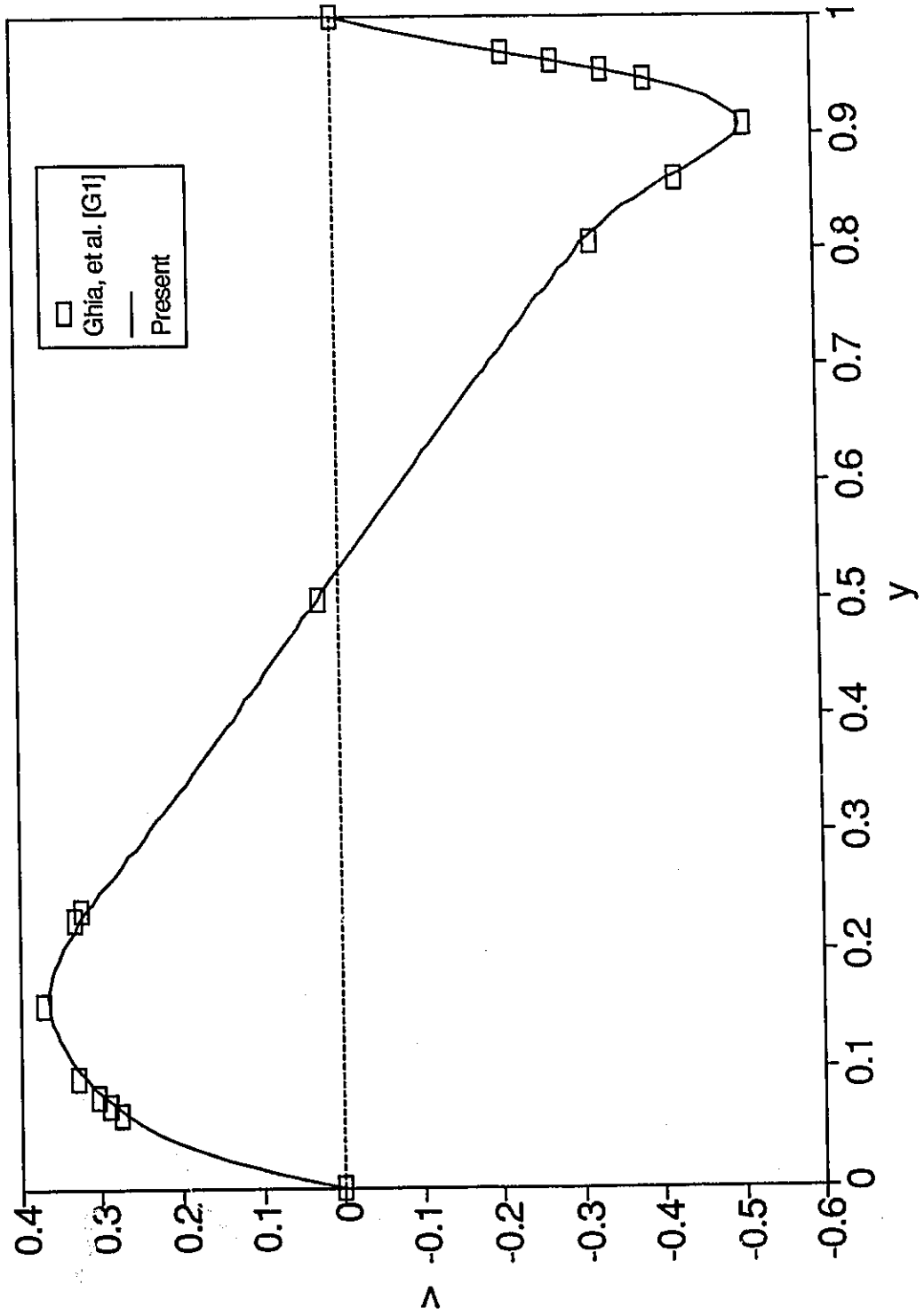

Figure 4.4 v-component of Velocity Along the Line y = 0.5 for Re = 1000

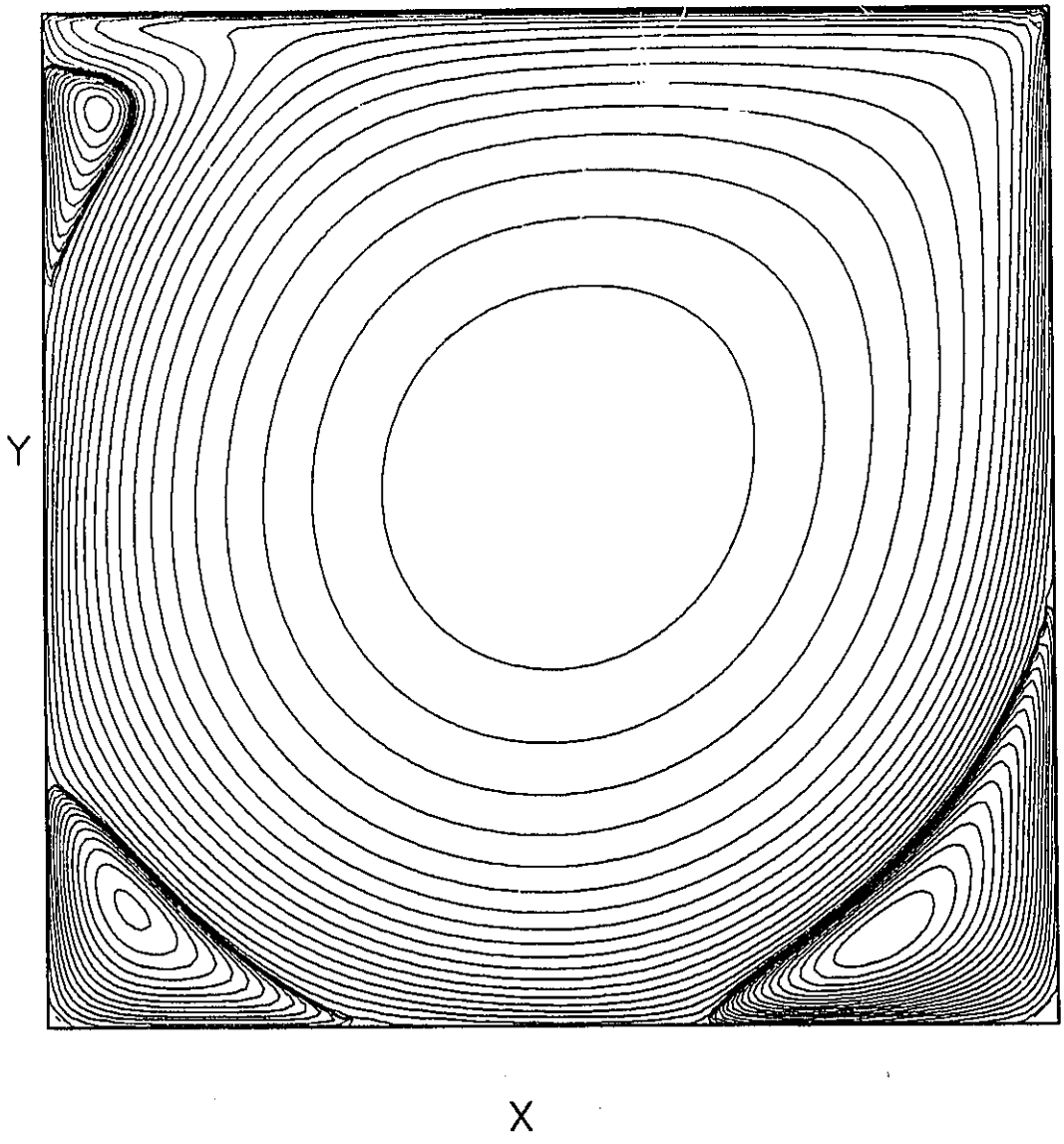

Figure 4.5 Cavity Flow Stream Function Contours for Re = 3200 on a 129x129 Uniform Grid

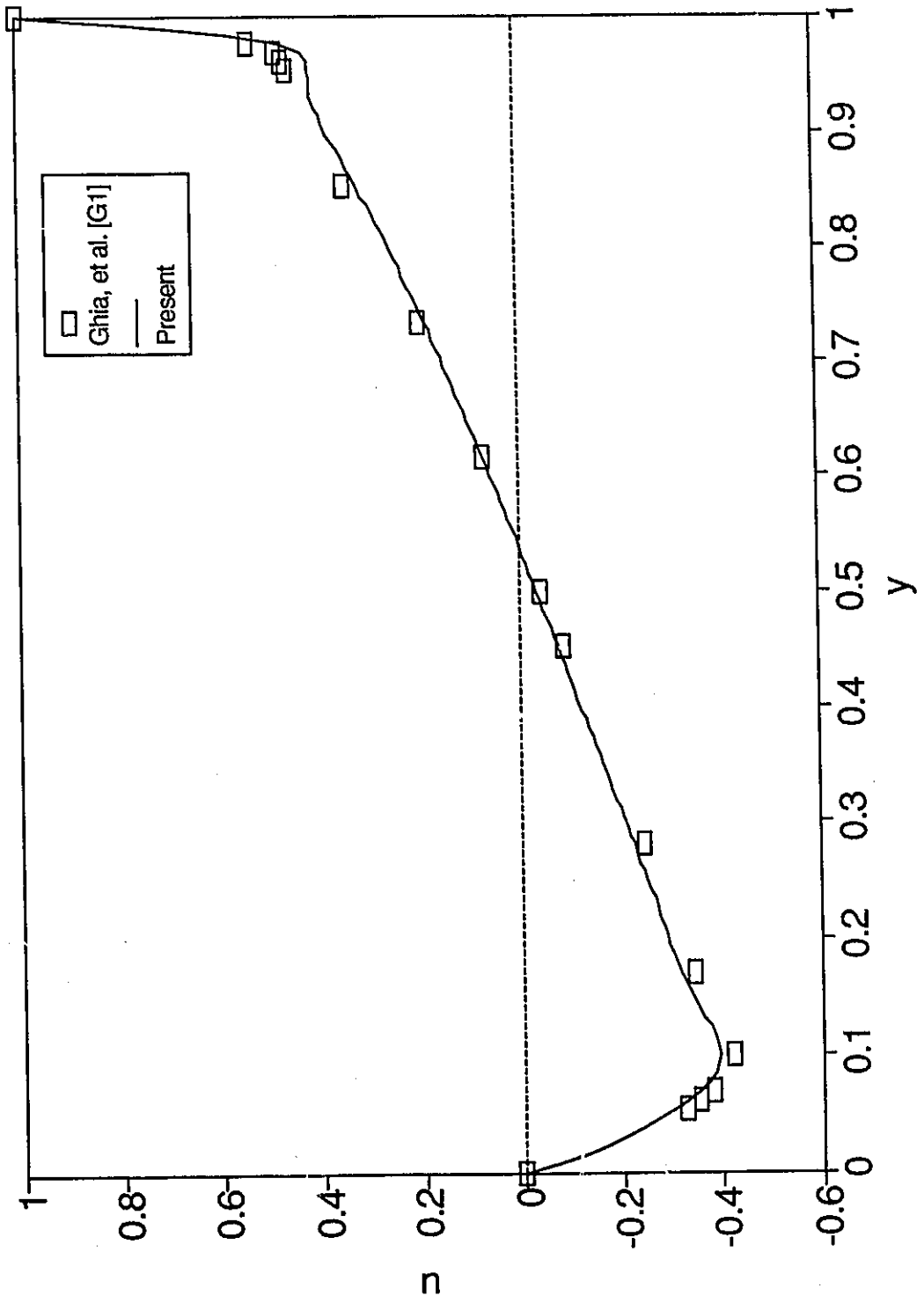

Figure 4.6 u-component of Velocity Along the Line x = 0.5 for Re = 3200

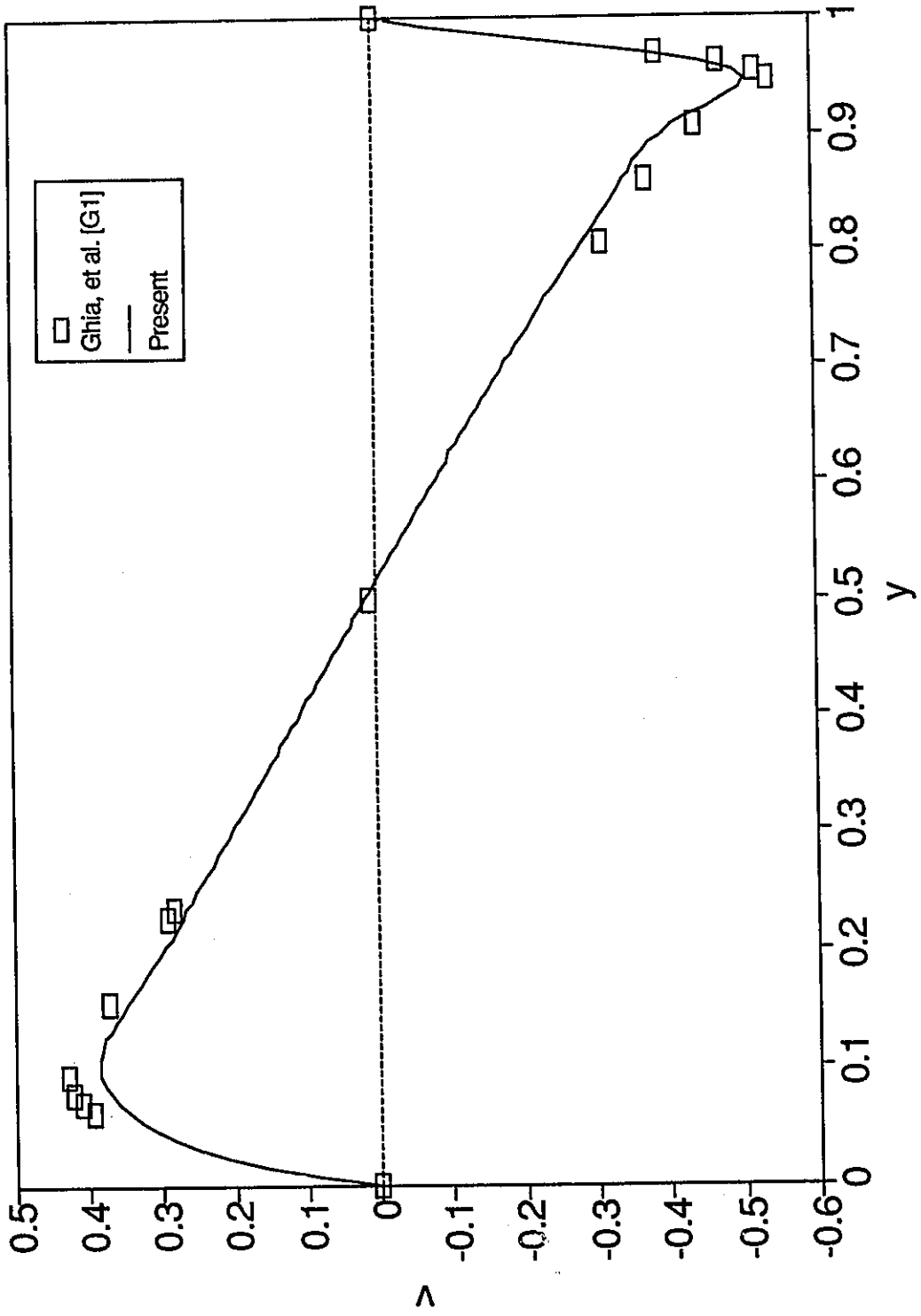

Figure 4.7 v-component of Velocity Along the Line y = 0.5 for Re = 3200

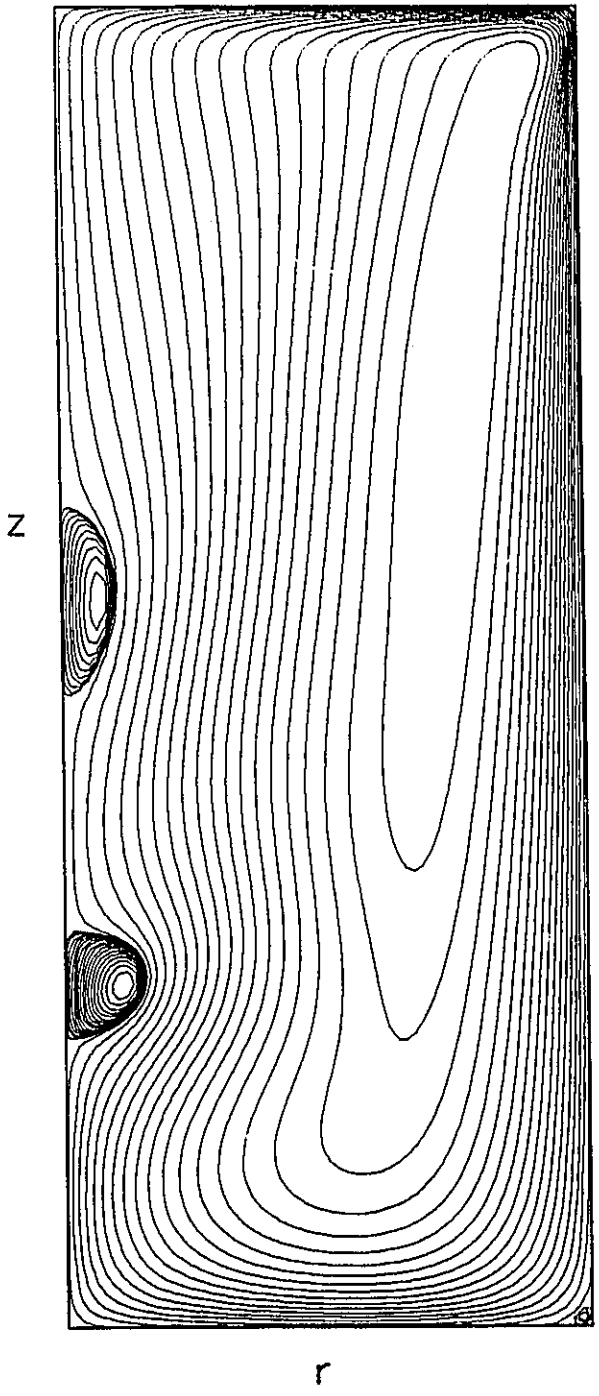

Figure 4.8 Rotating Flow Stream Function Contours for Re = 1994 and δ = 2.50 on a 61x151 Uniform Grid

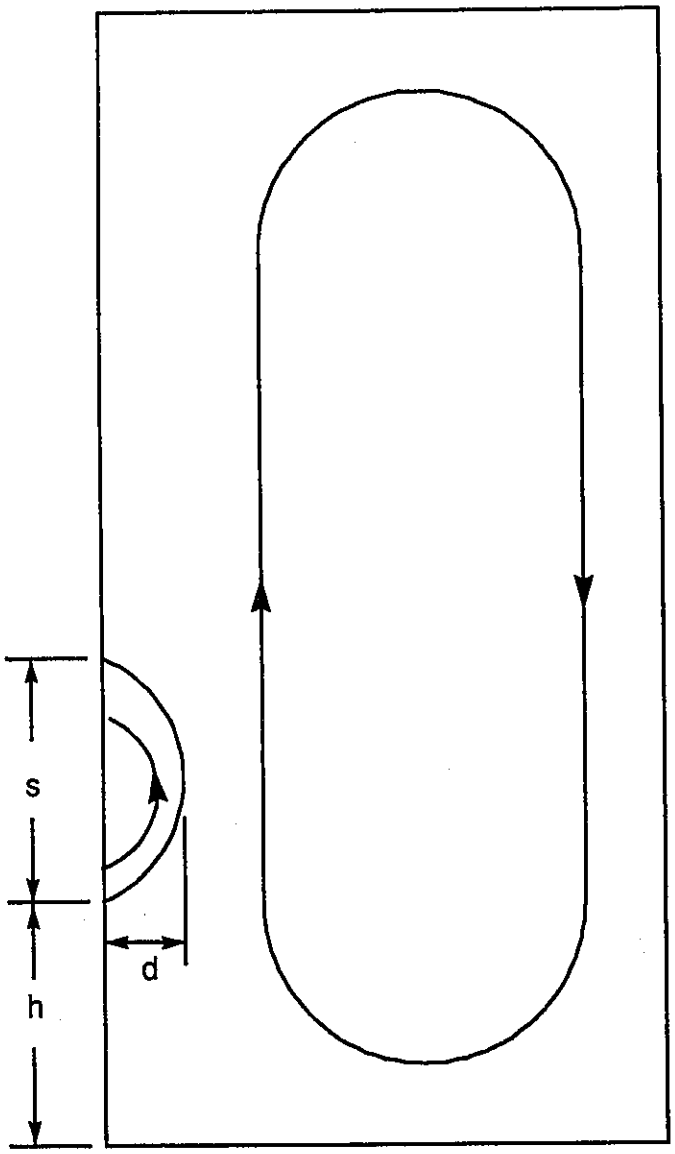

Figure 4.9 Nomenclature for the Dimensions of the Rotating Flow Recirculation Regions

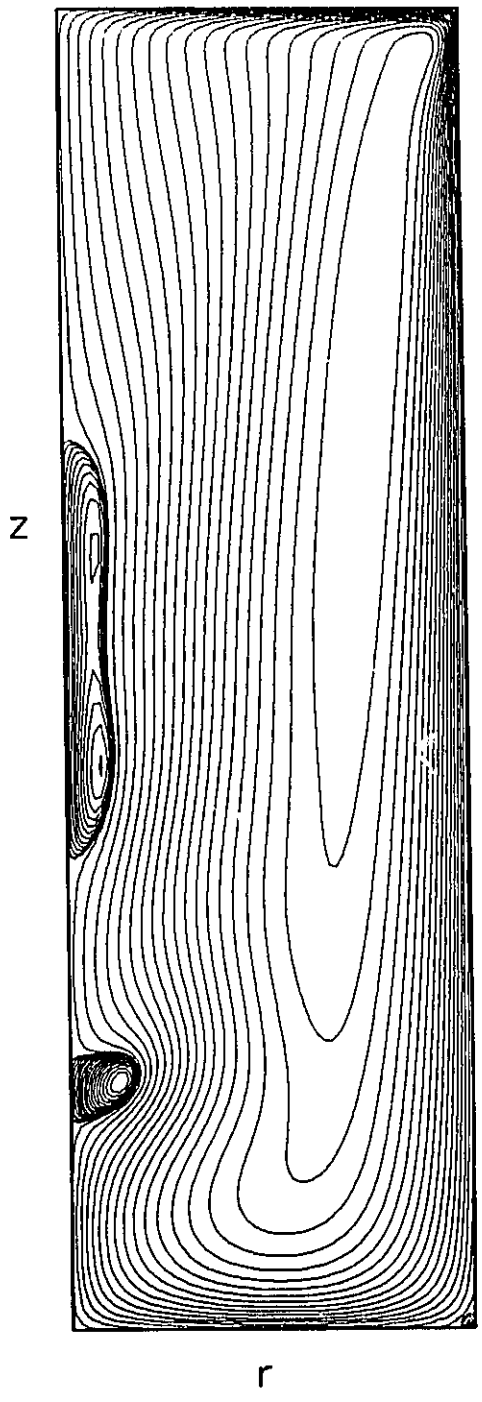

Figure 4.10 Rotating Flow Stream Function Contours for Re = 2752 and δ = 3.25 on a 61x196 Uniform Grid

Figure 4.11 Legend for Figures 4.12 to 4.23

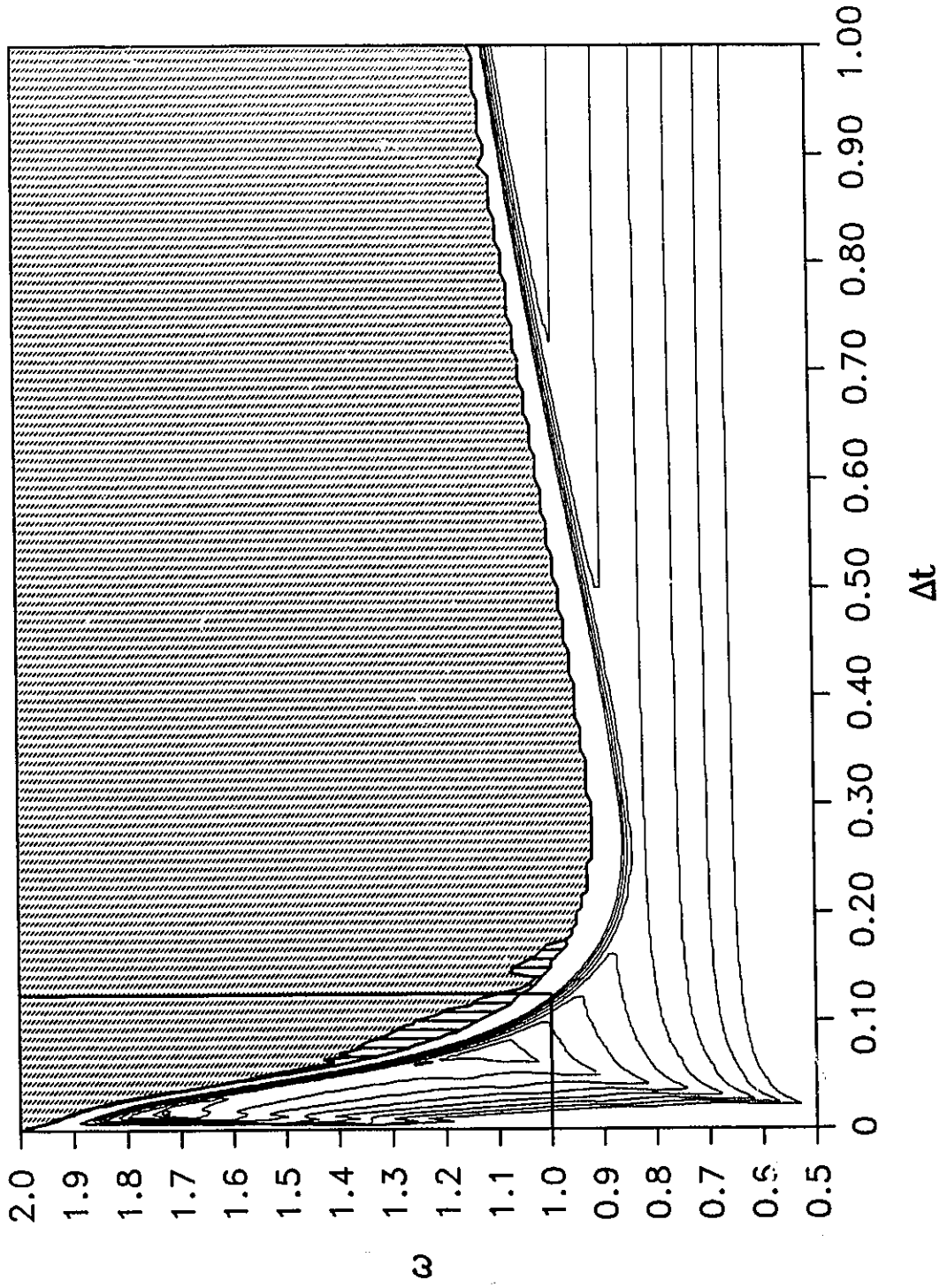

Figure 4.12 Iteration Contour Graph for Re = 10 on a 21x21 Grid (see Figure 4.11 for legend)

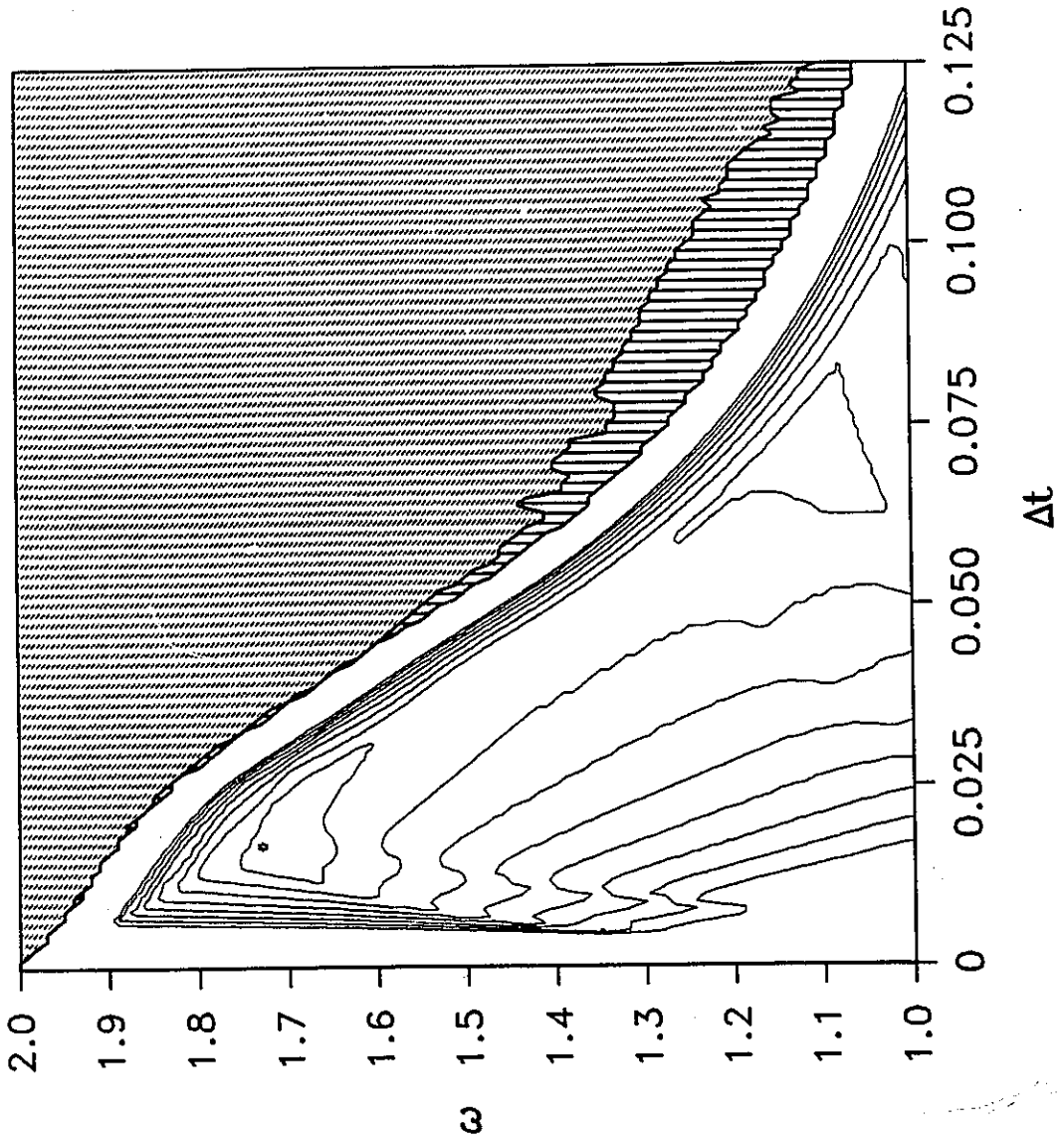

Figure 4.13 Iteration Contour Graph for Re = 10 on a 21x21 Grid (see insert of Figure 4.12)

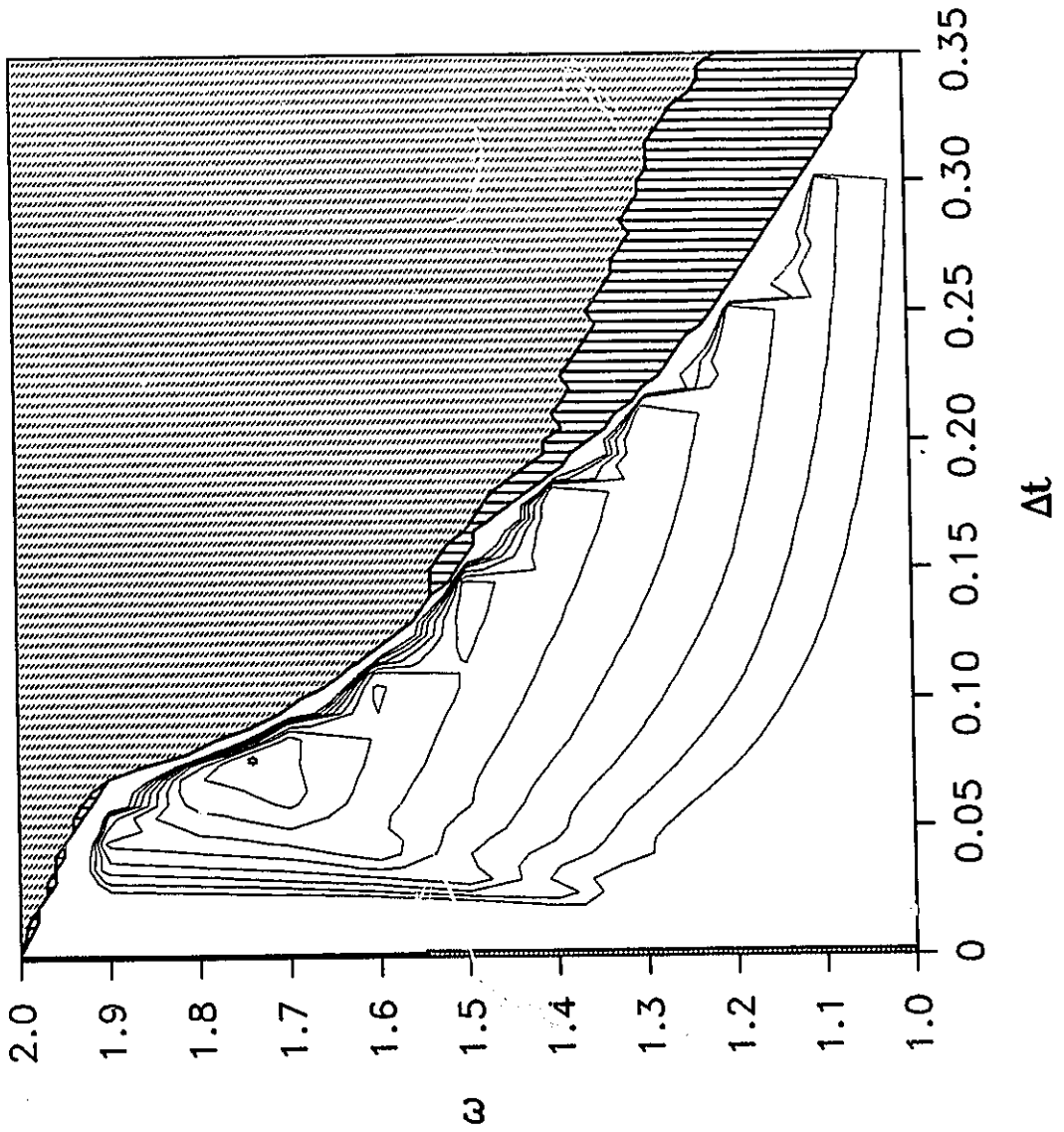

Figure 4.14 Iteration Contour Graph for Re = 100 on a 36x36 Grid (see Figure 4.11 for legend)

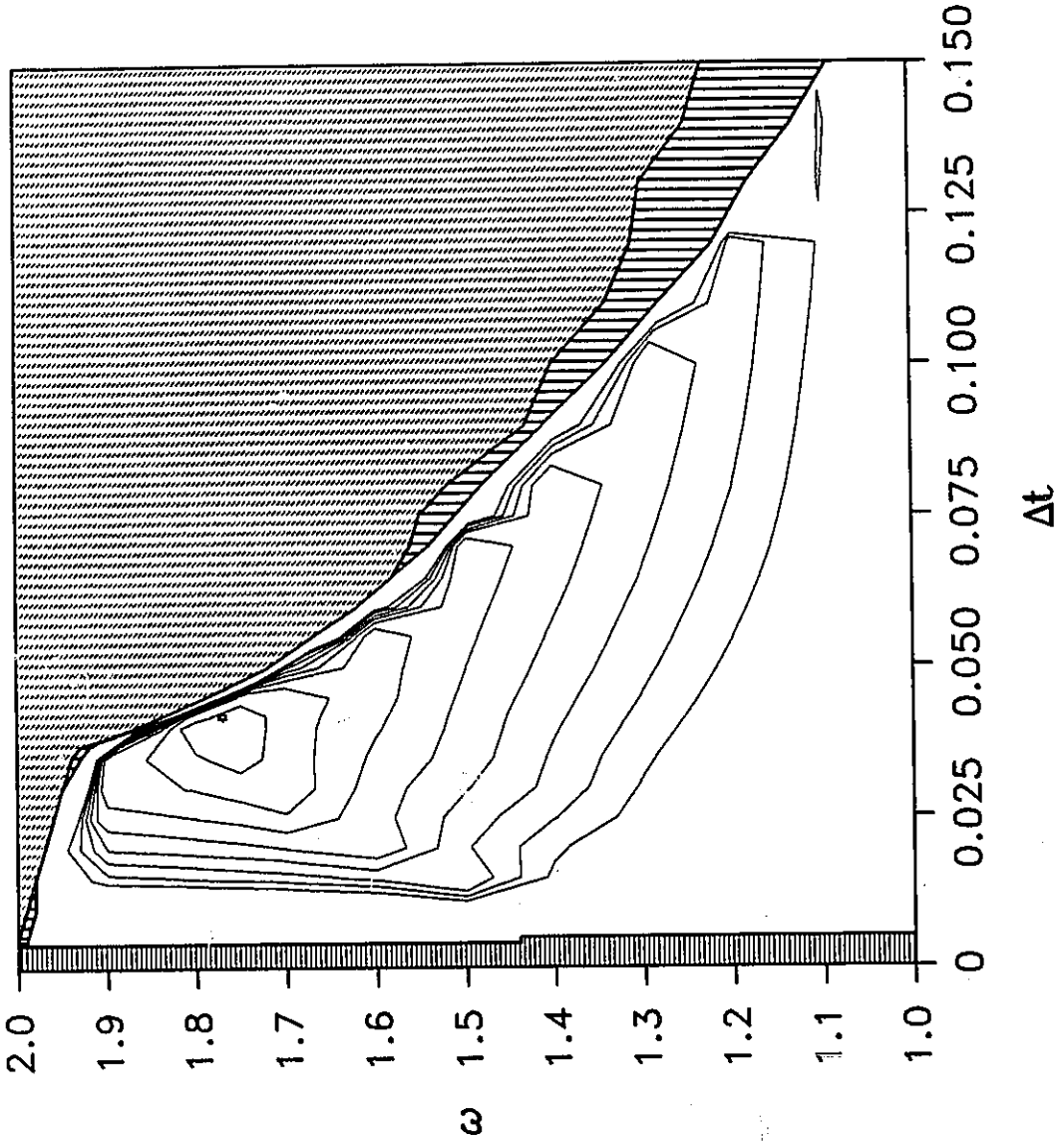

Figure 4.15 Iteration Contour Graph for Re = 100 on a 51x51 Grid (see Figure 4.11 for legend)

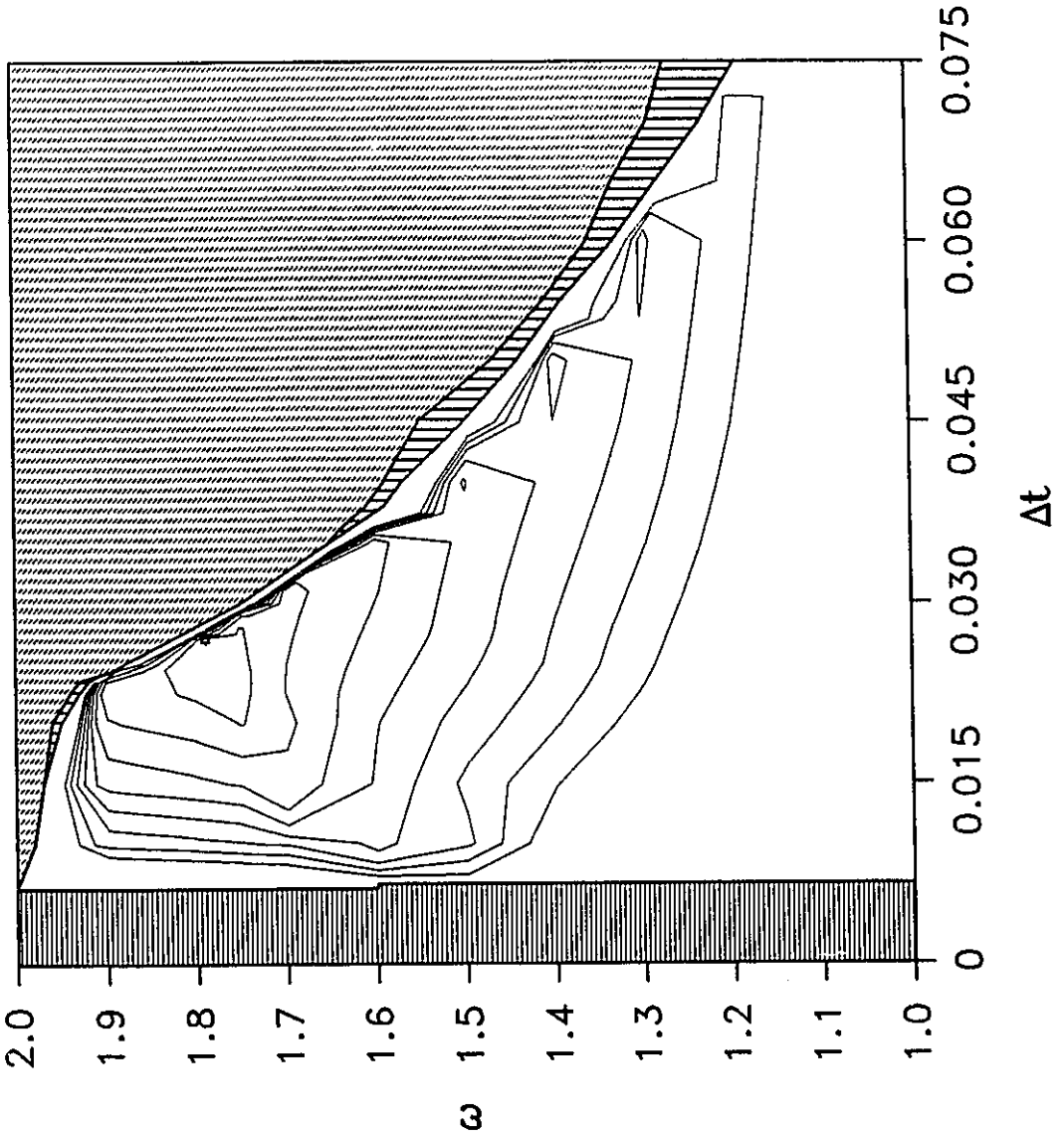

Figure 4.16 Iteration Contour Graph for Re = 100 on a 66x66 Grid (see Figure 4.11 for legend)

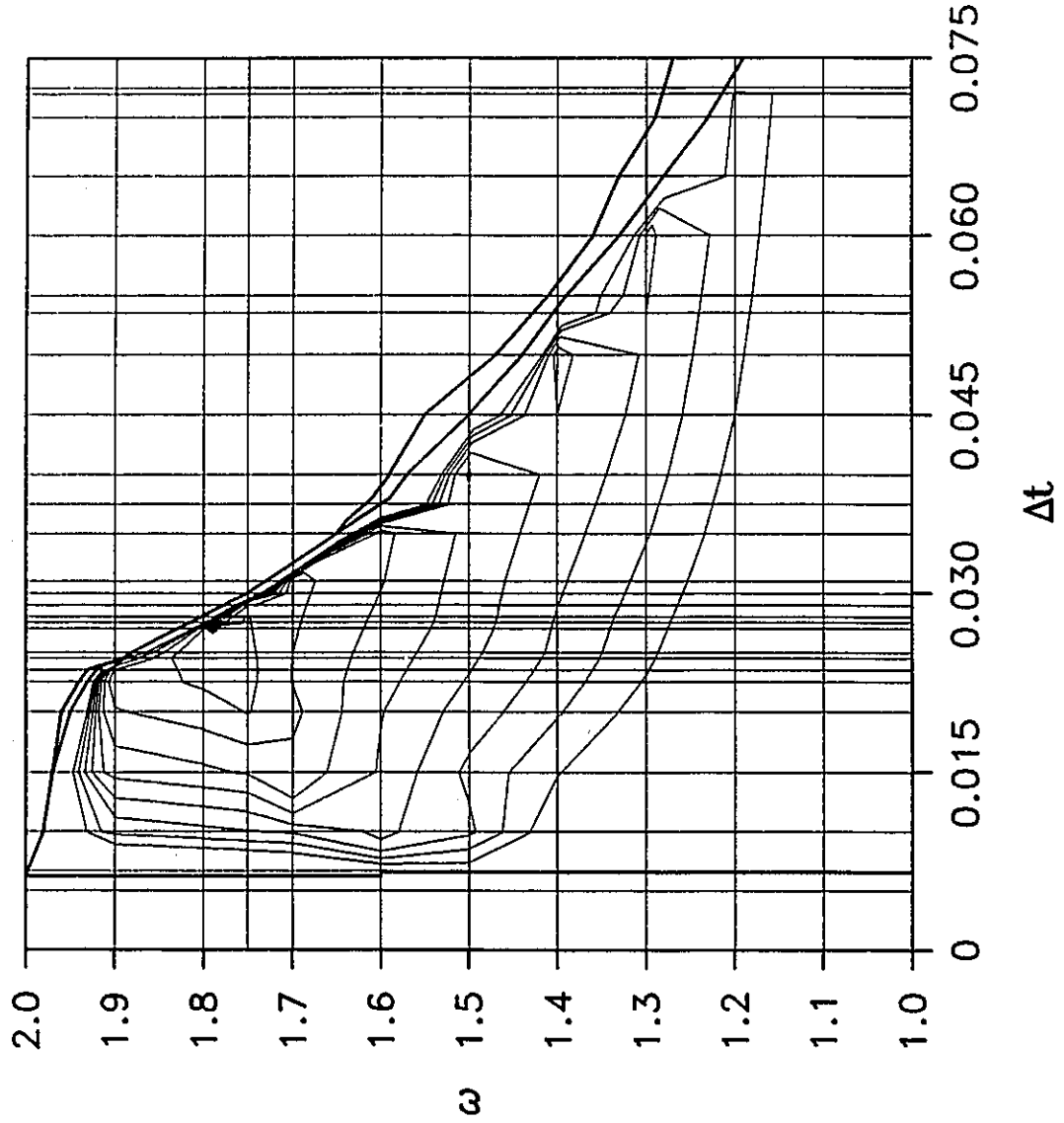

Figure 4.17 Mesh for ω-Δt of Figure 4.16

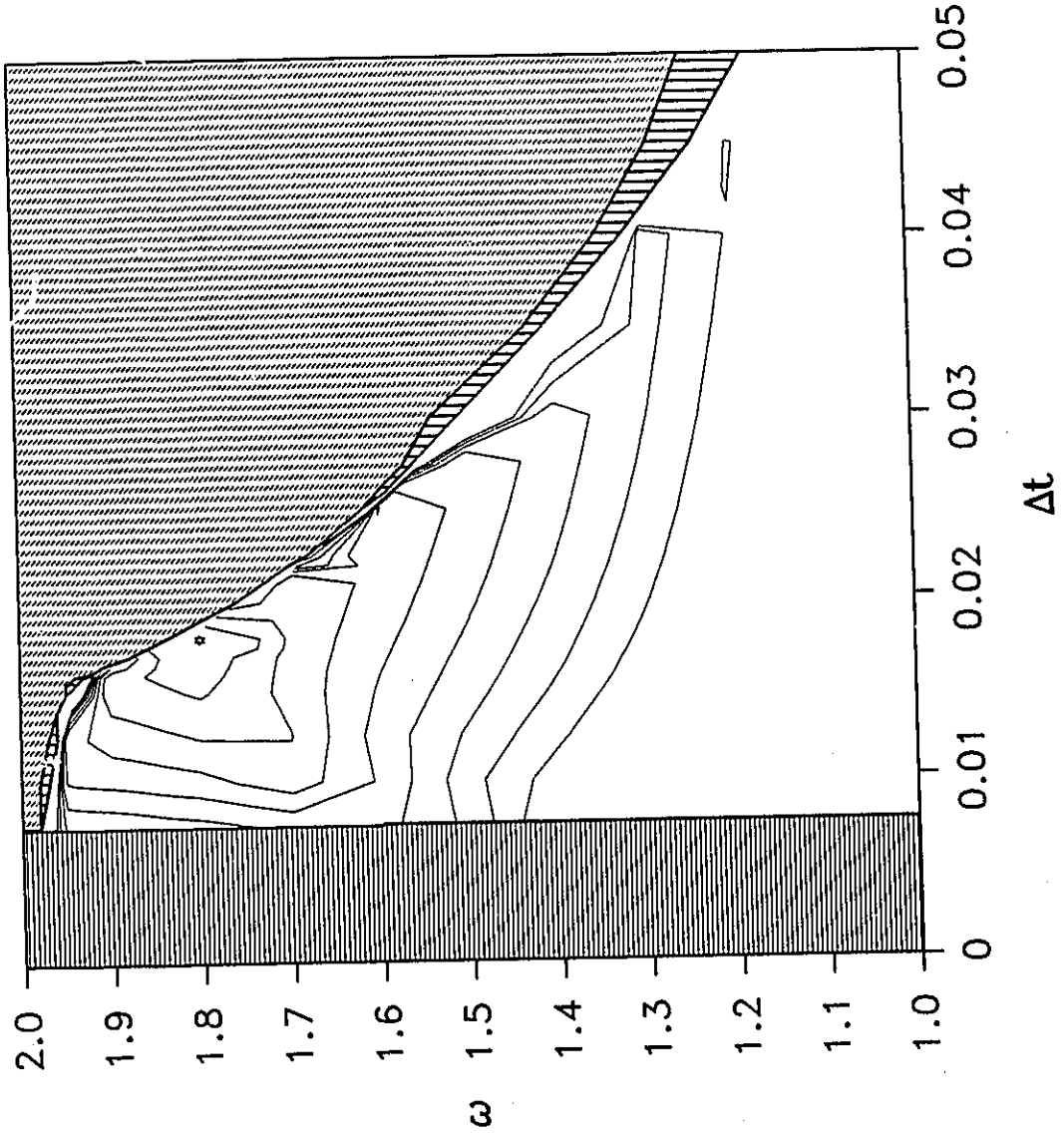

Figure 4.18 Iteration Contour Graph for Re = 100 on an 81x81 Grid (see Figure 4.11 for legend)

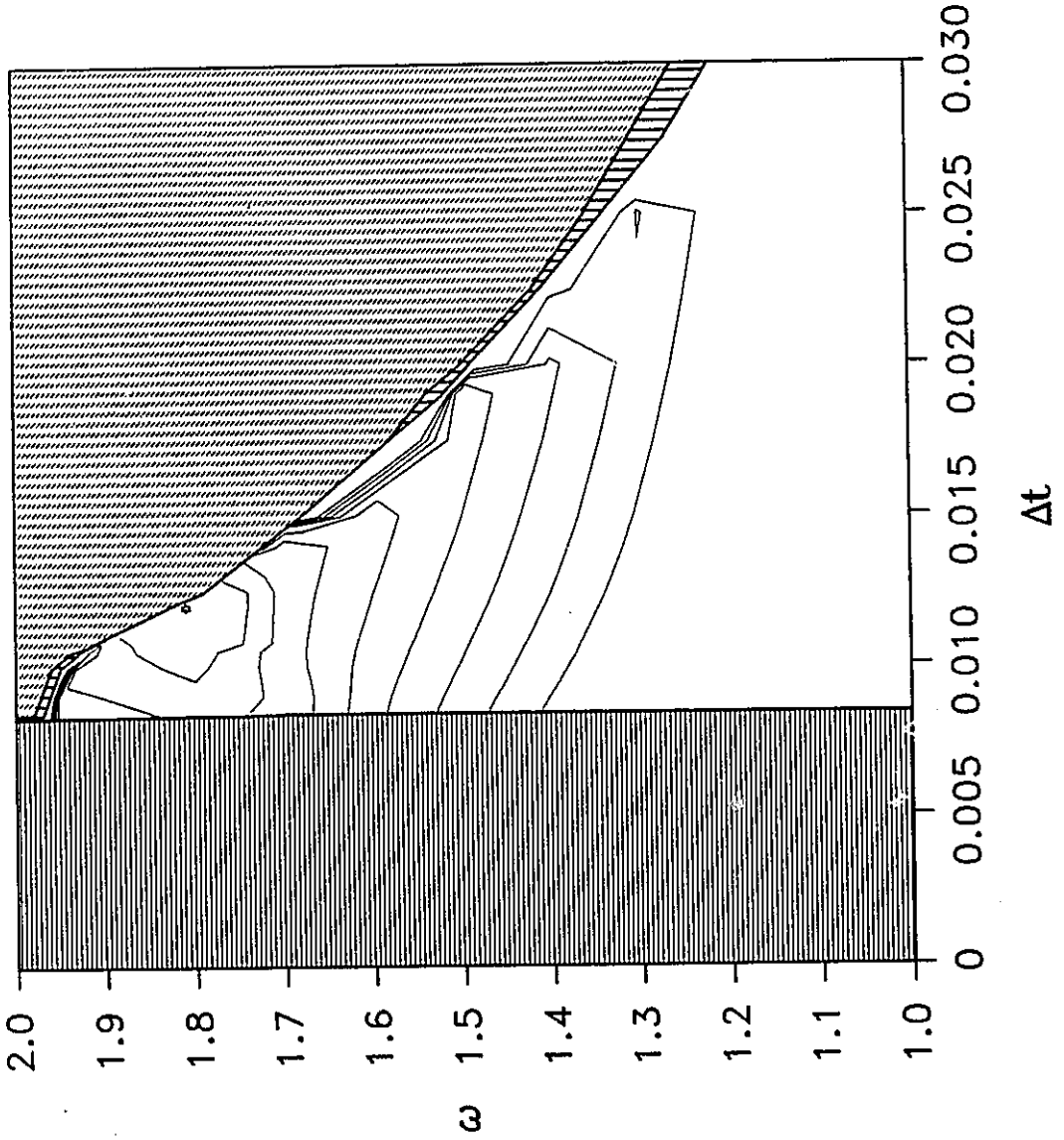

Figure 4.19 Iteration Contour Graph for Re = 100 on a 101x101 Grid (see Figure 4.11 for legend)

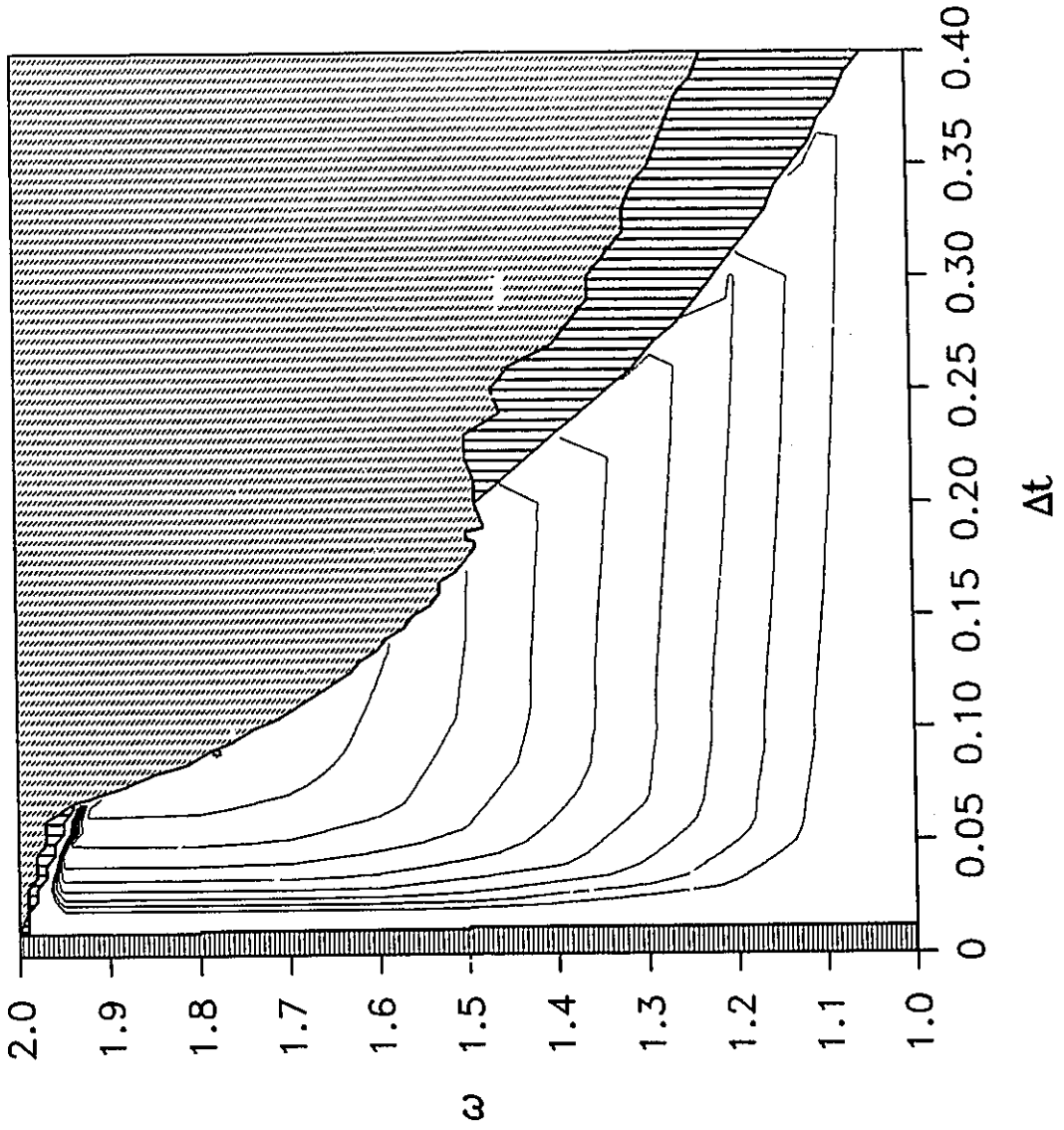

Figure 4.20 Iteration Contour Graph for Re = 250 on a 51x51 Grid (see Figure 4.11 for legend)

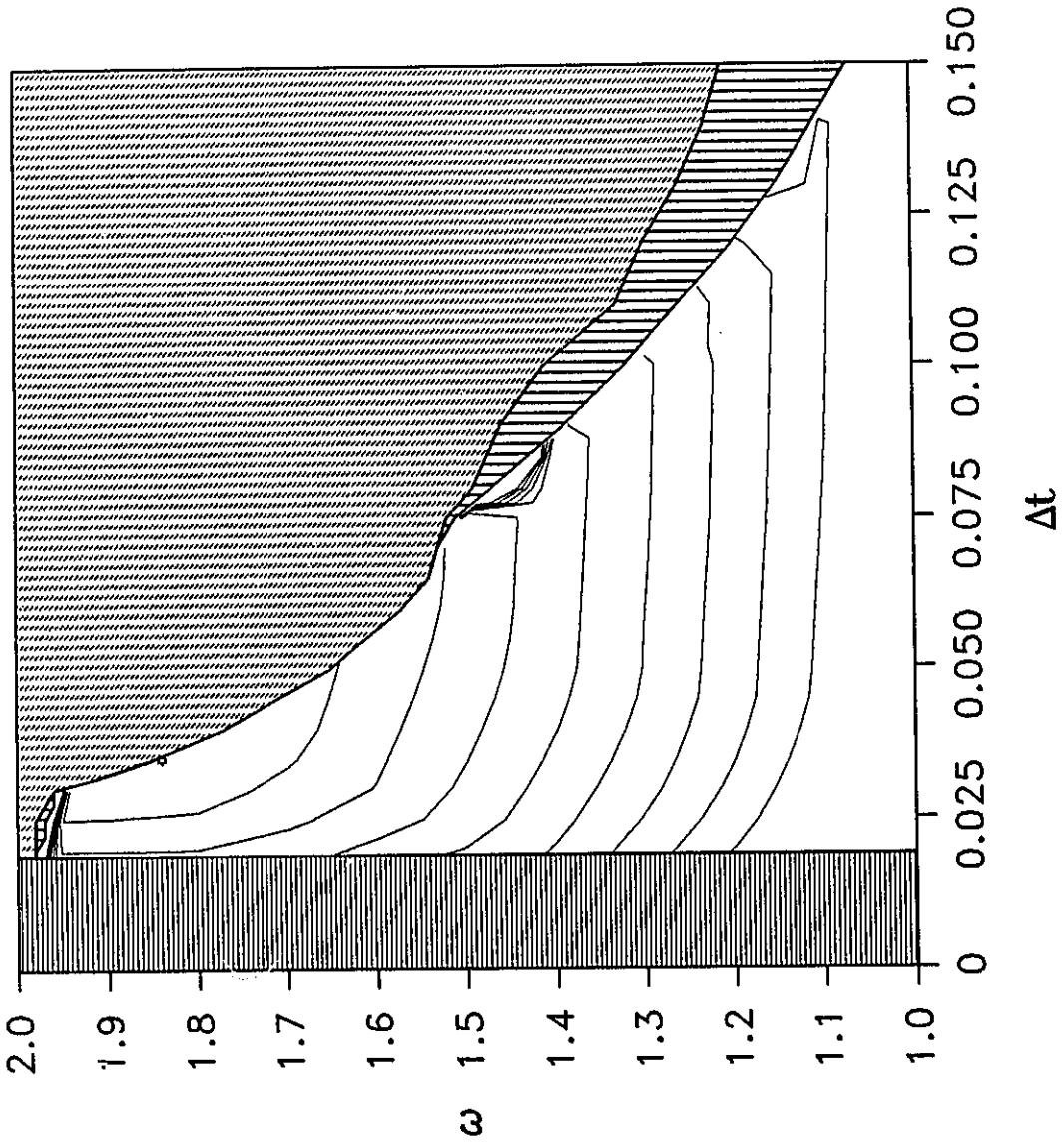

Figure 4.21 Iteration Contour Graph for Re = 250 on an 81x81 Grid (see Figure 4.11 for legend)

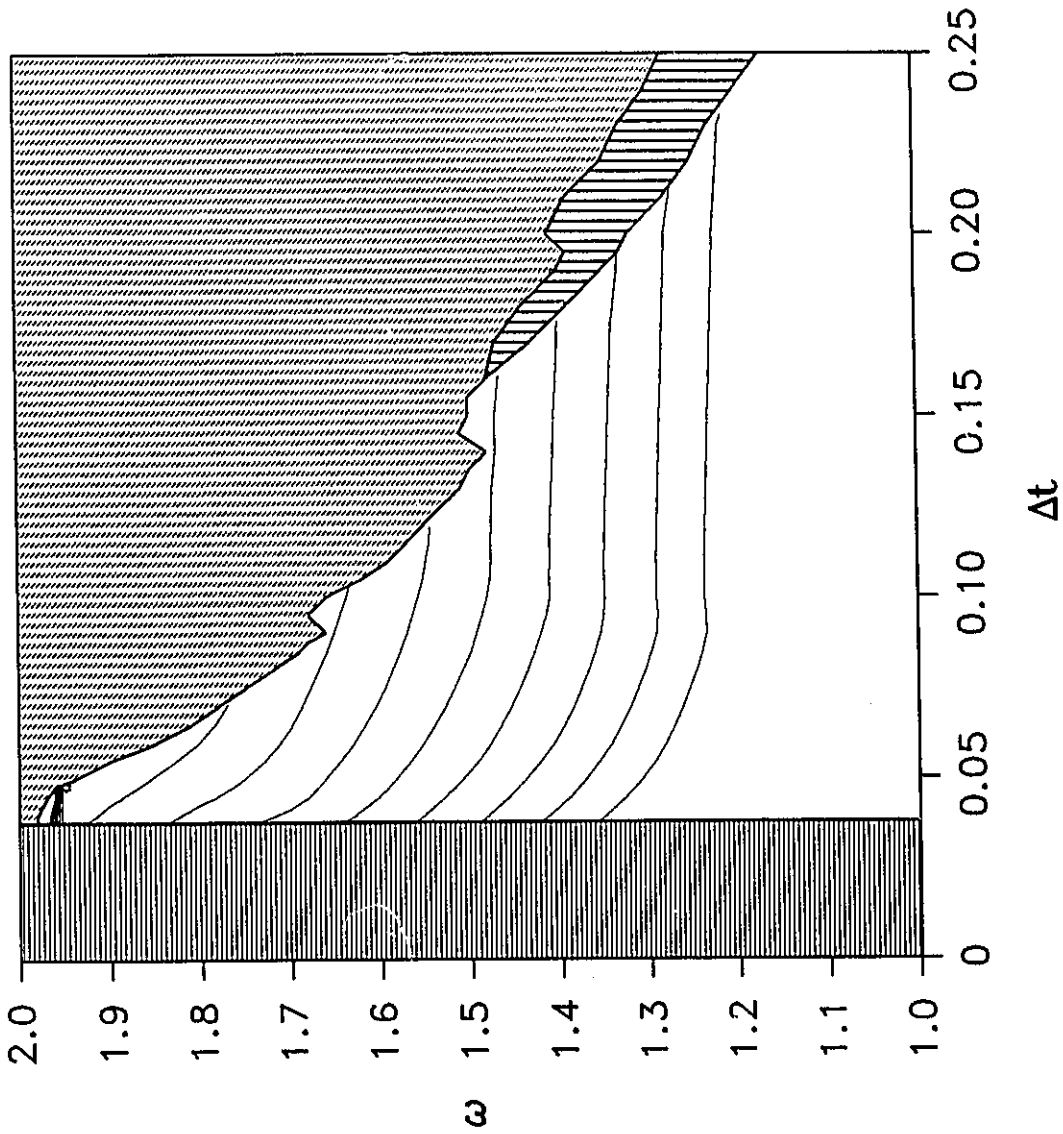

Figure 4.22 Iteration Contour Graph for Re = 500 on an 81x81 Grid (see Figure 4.11 for legend)

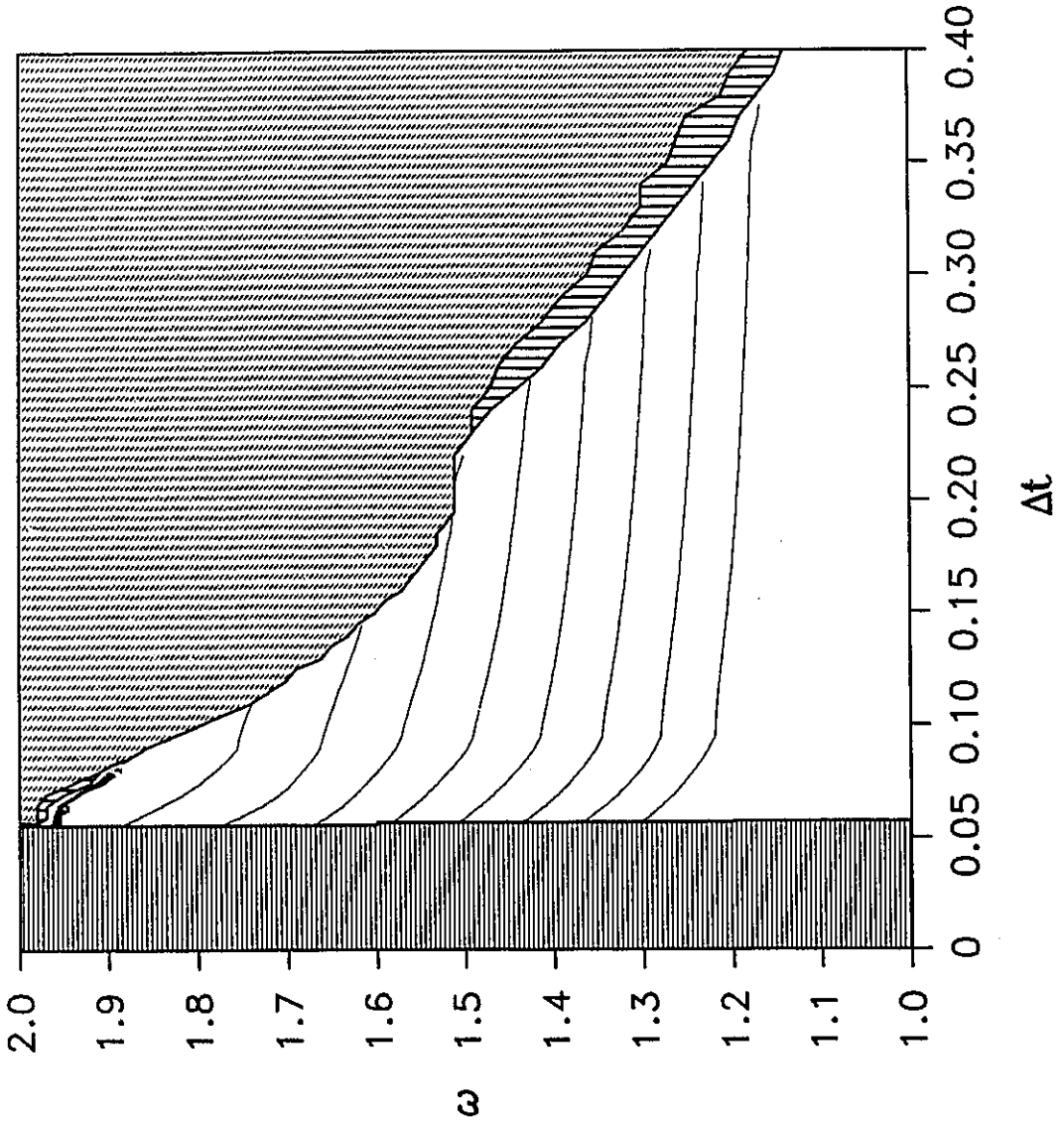

Figure 4.23 Iteration Contour Graph for Re = 750 on an 81x81 Grid (see Figure 4.11 for legend)

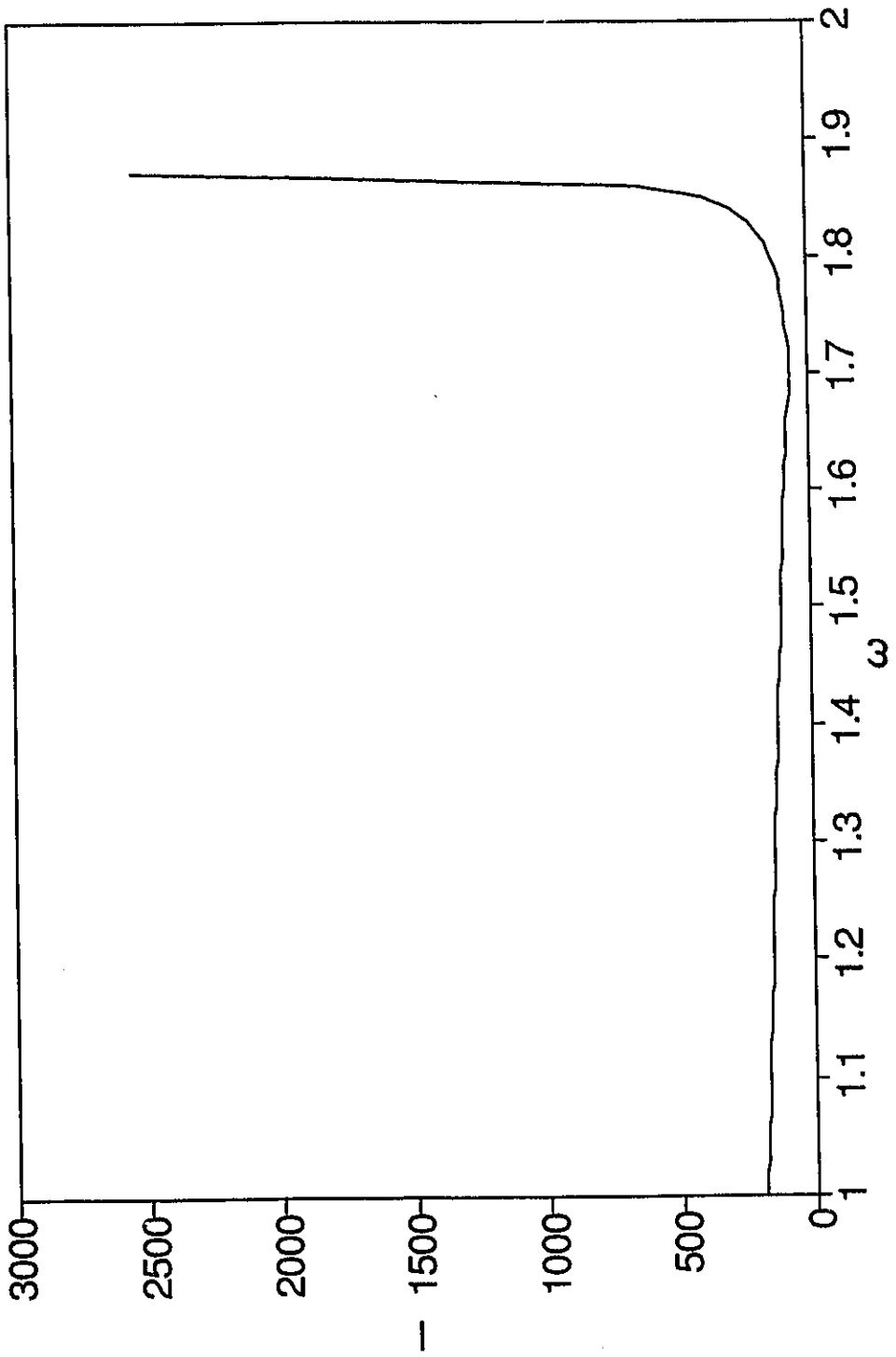

Figure 4.24 Cross Section through Figure 4.13 at $\Delta t = \Delta t_{opt} = 0.019$ (Re = 10 on a 21x21 Grid)

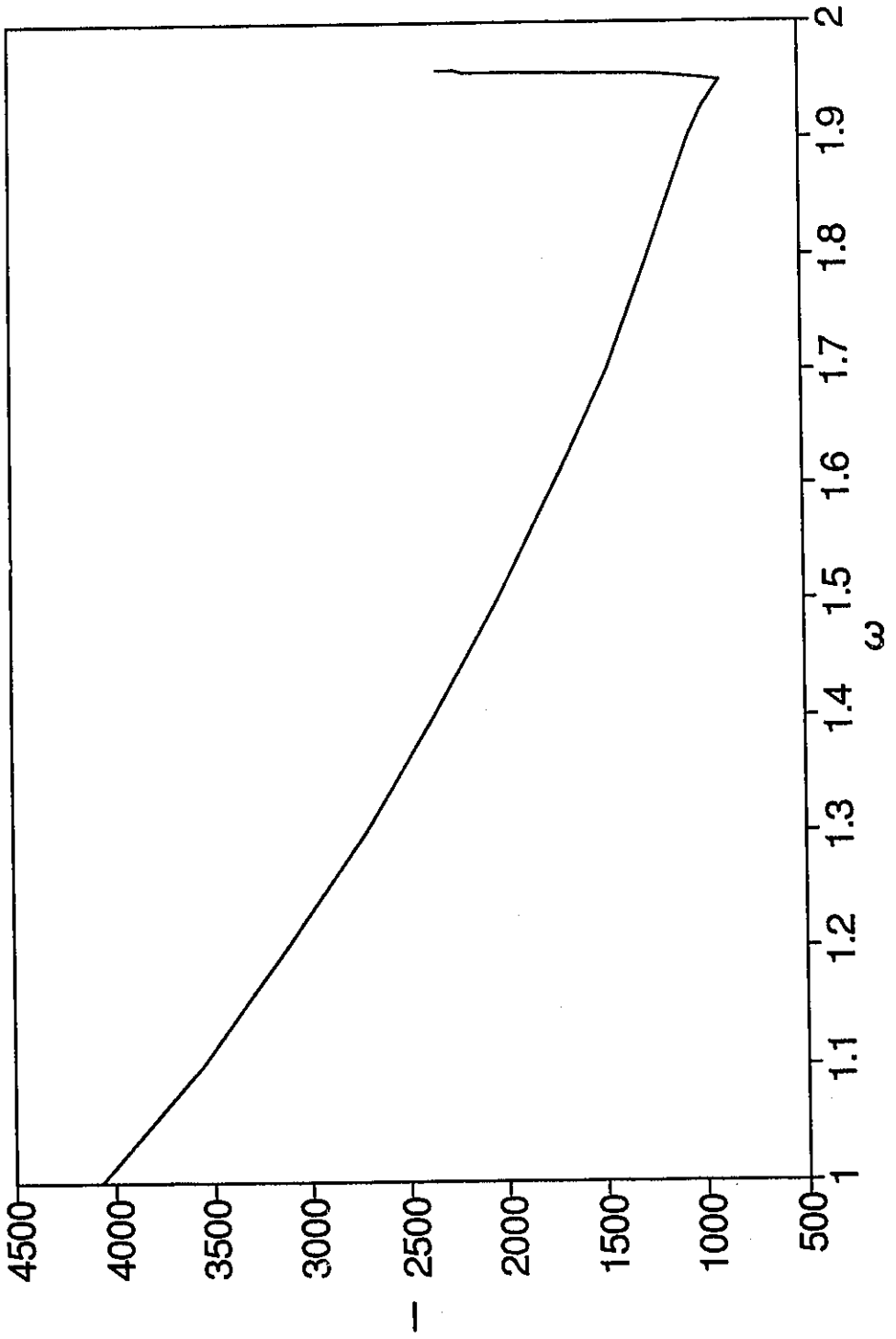

Figure 4.25 Cross Section through Figure 4.22 at $\Delta t = \Delta t_{opt} = 0.048$ (Re = 500 on an 81x81 Grid)

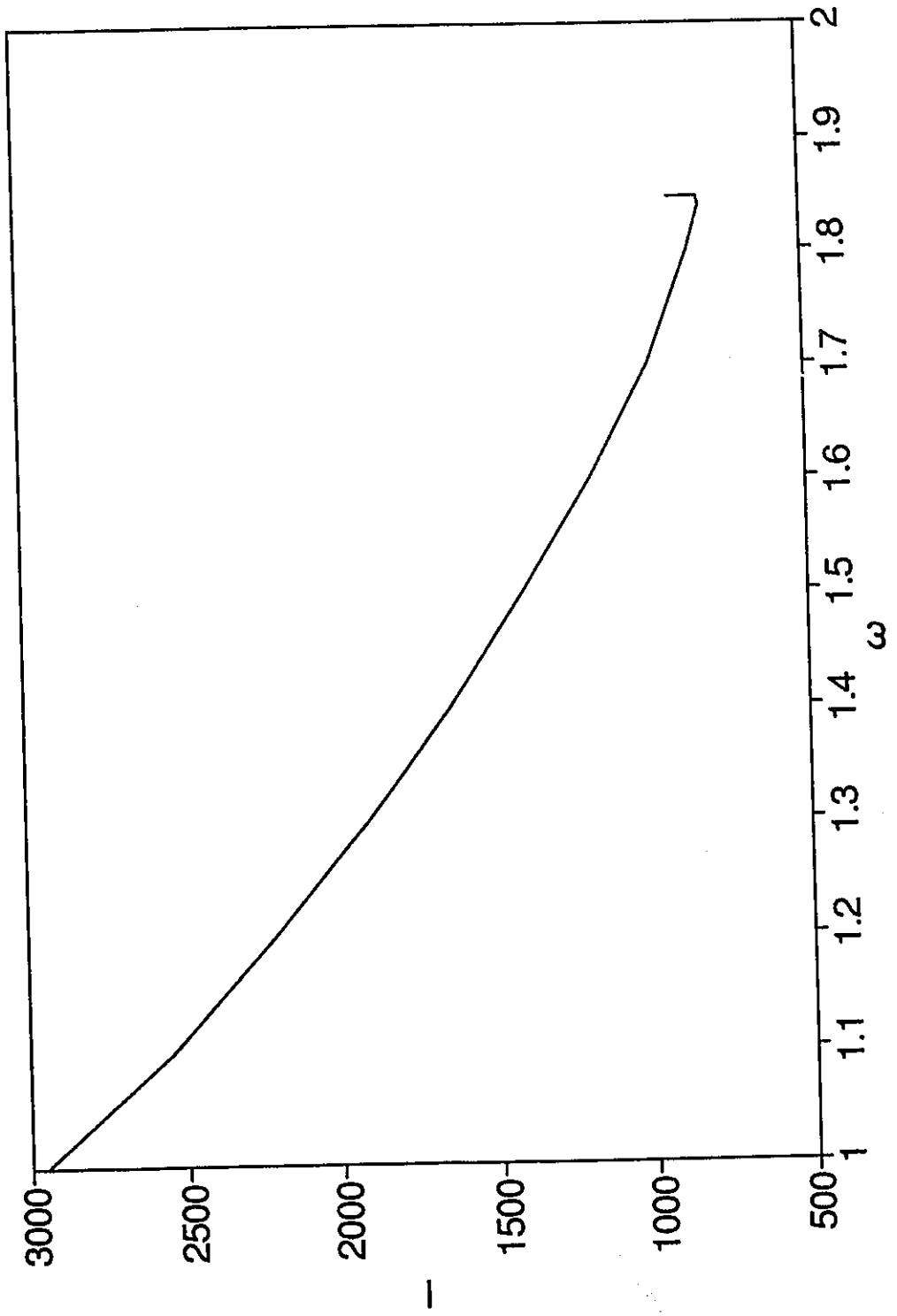

Figure 4.26 Cross Section through Figure 4.21 at $\Delta t = \Delta t_{opt} = 0.035$ (Re = 250 on an 81x81 Grid)

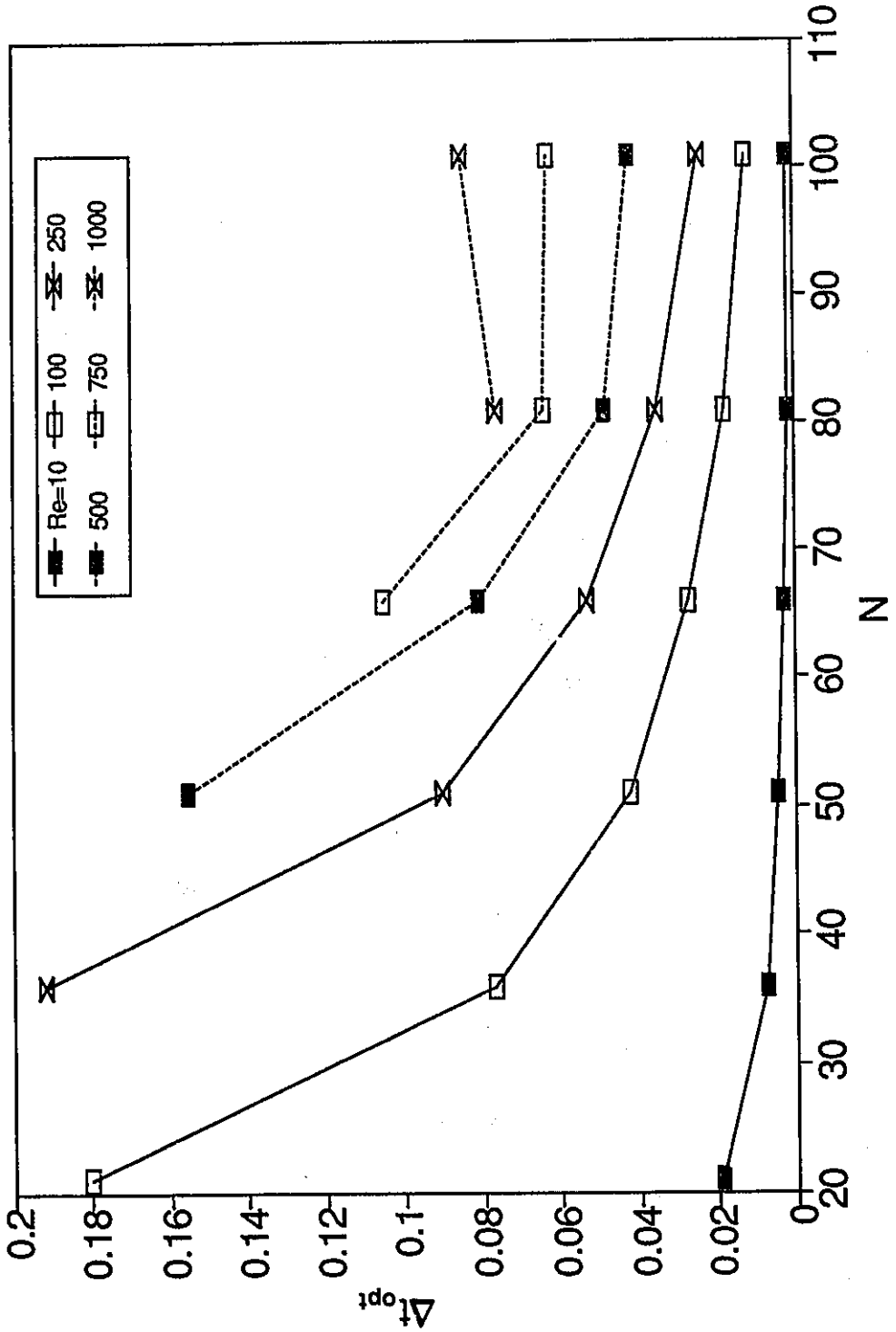

Figure 4.27 Δt_{opt} as a Function of the Number of Nodes in the x-direction for Various Reynolds Numbers

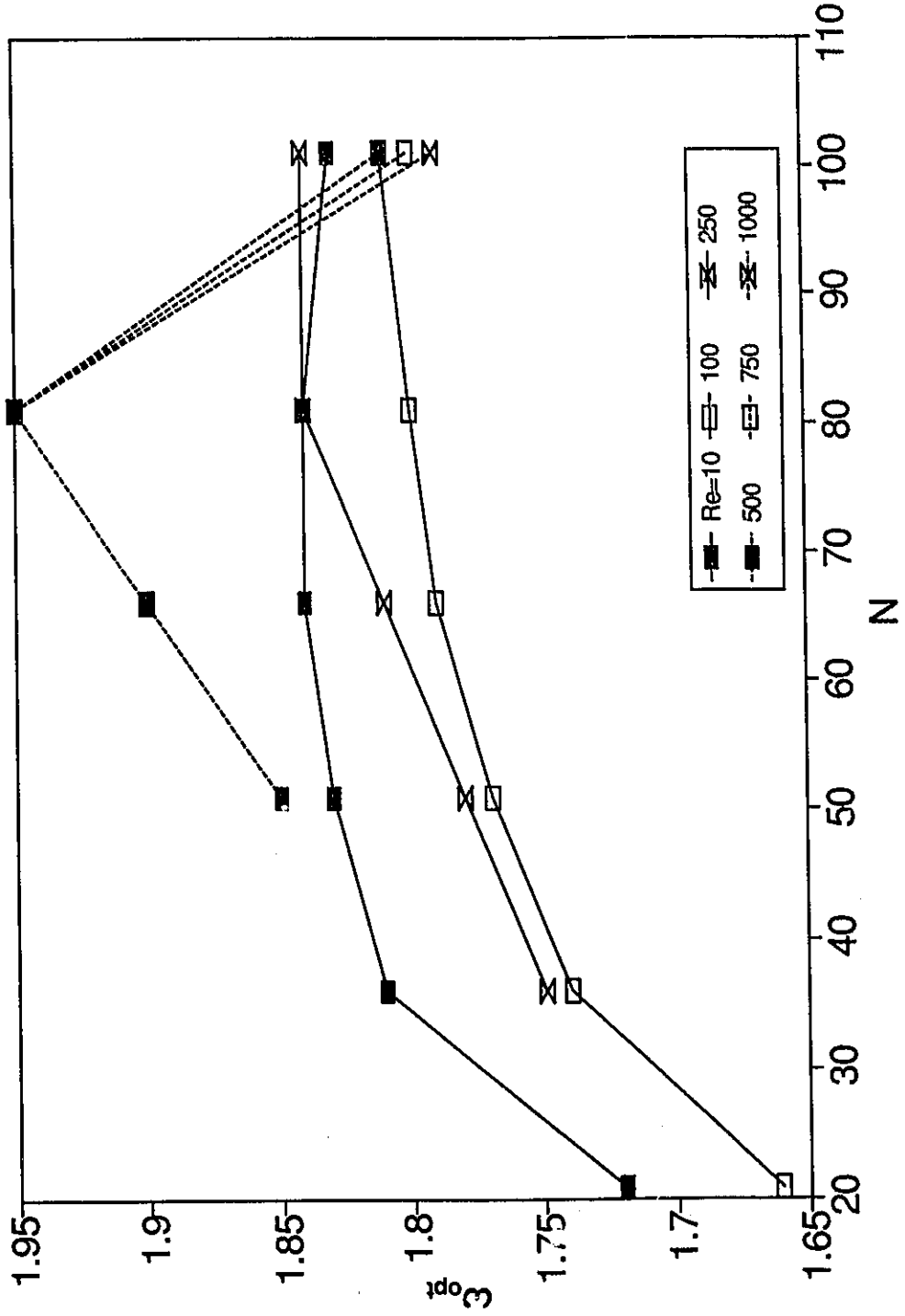

Figure 4.28 ω_{opt} as a Function of the Number of Nodes in the x-direction for Various Reynolds Numbers

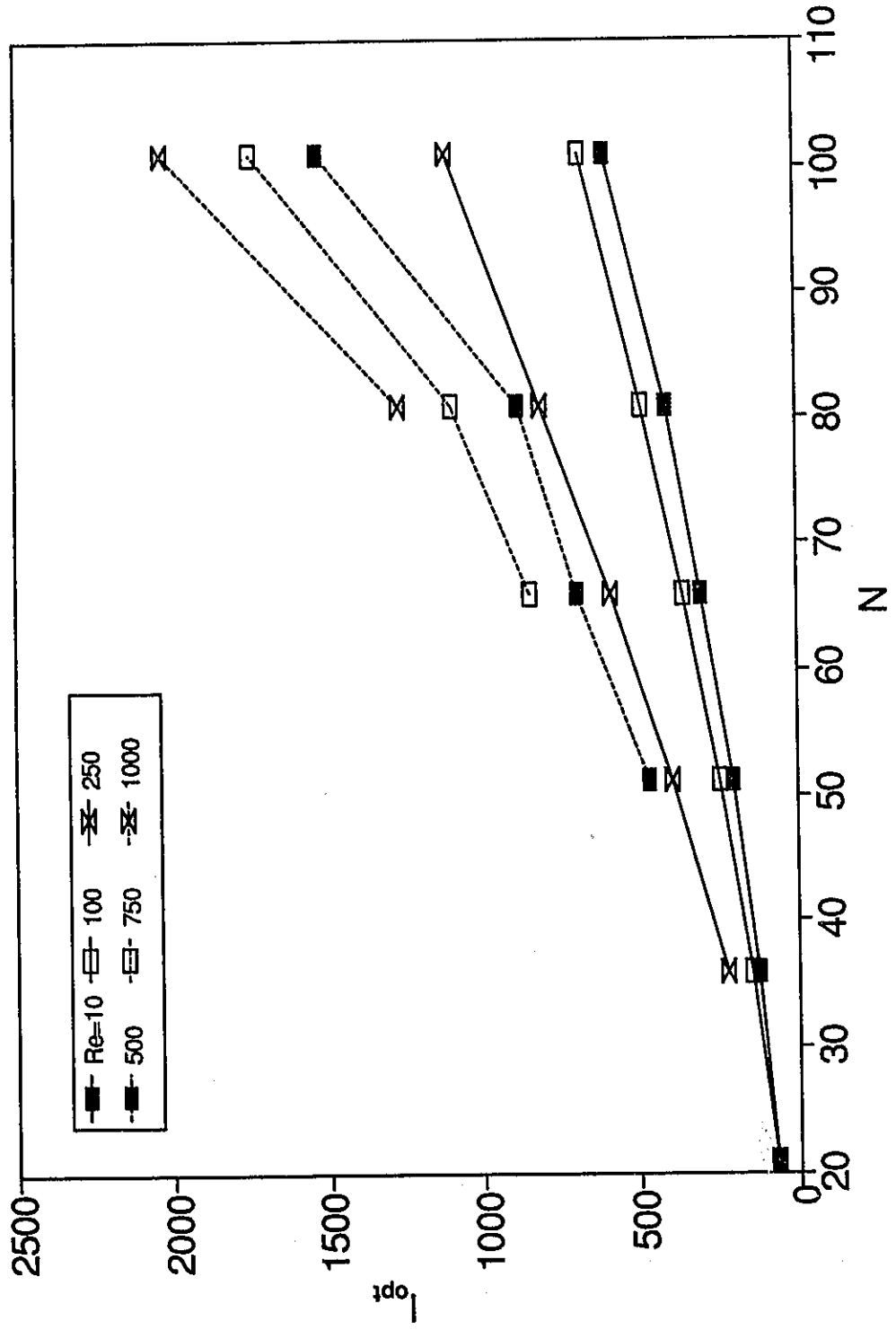

Figure 4.29 l_{opt} as a Function of the Number of Nodes in the x-direction for Various Reynolds Numbers

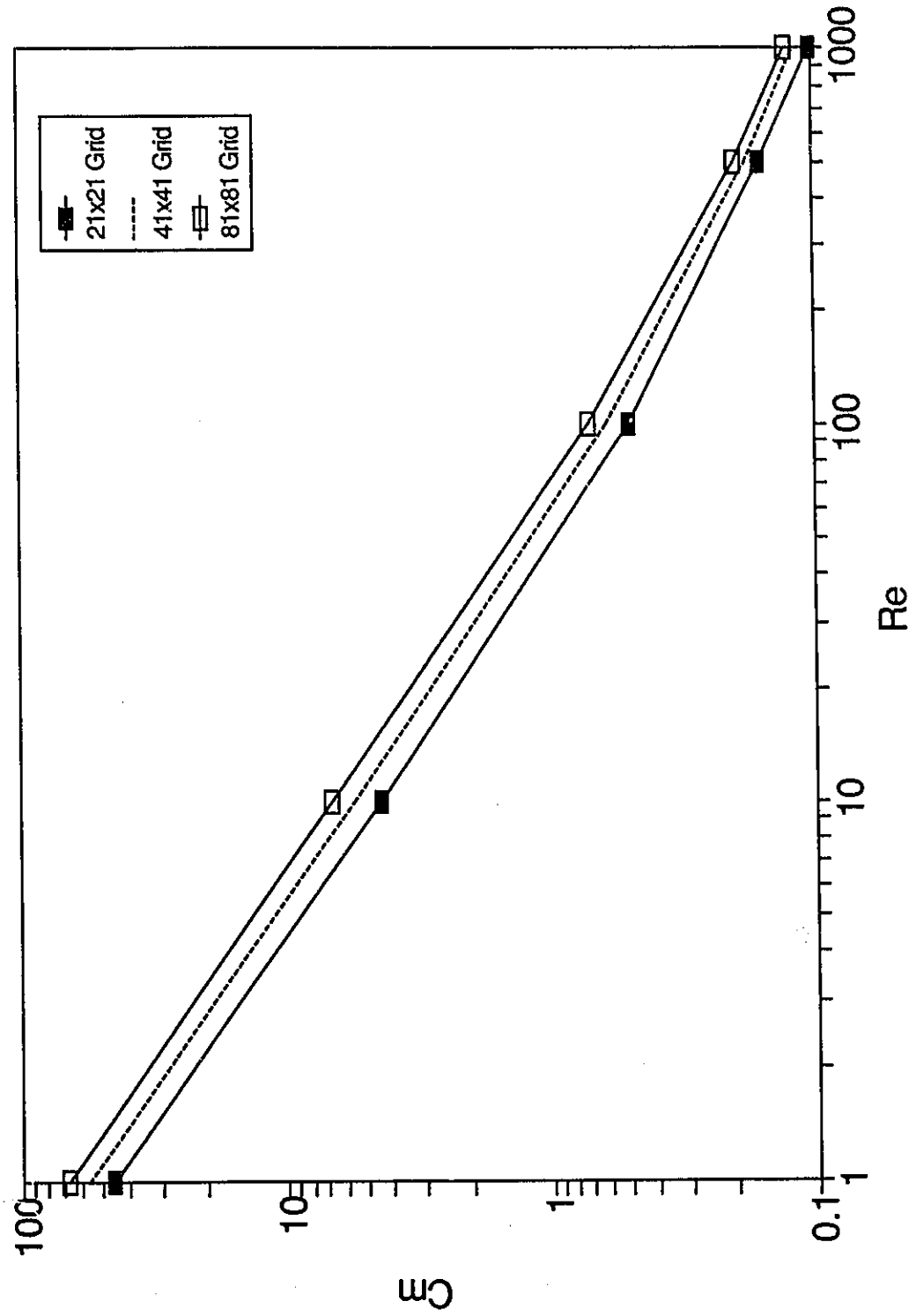

Figure 4.30 Torque Coefficient Calculated Using Pao's Method

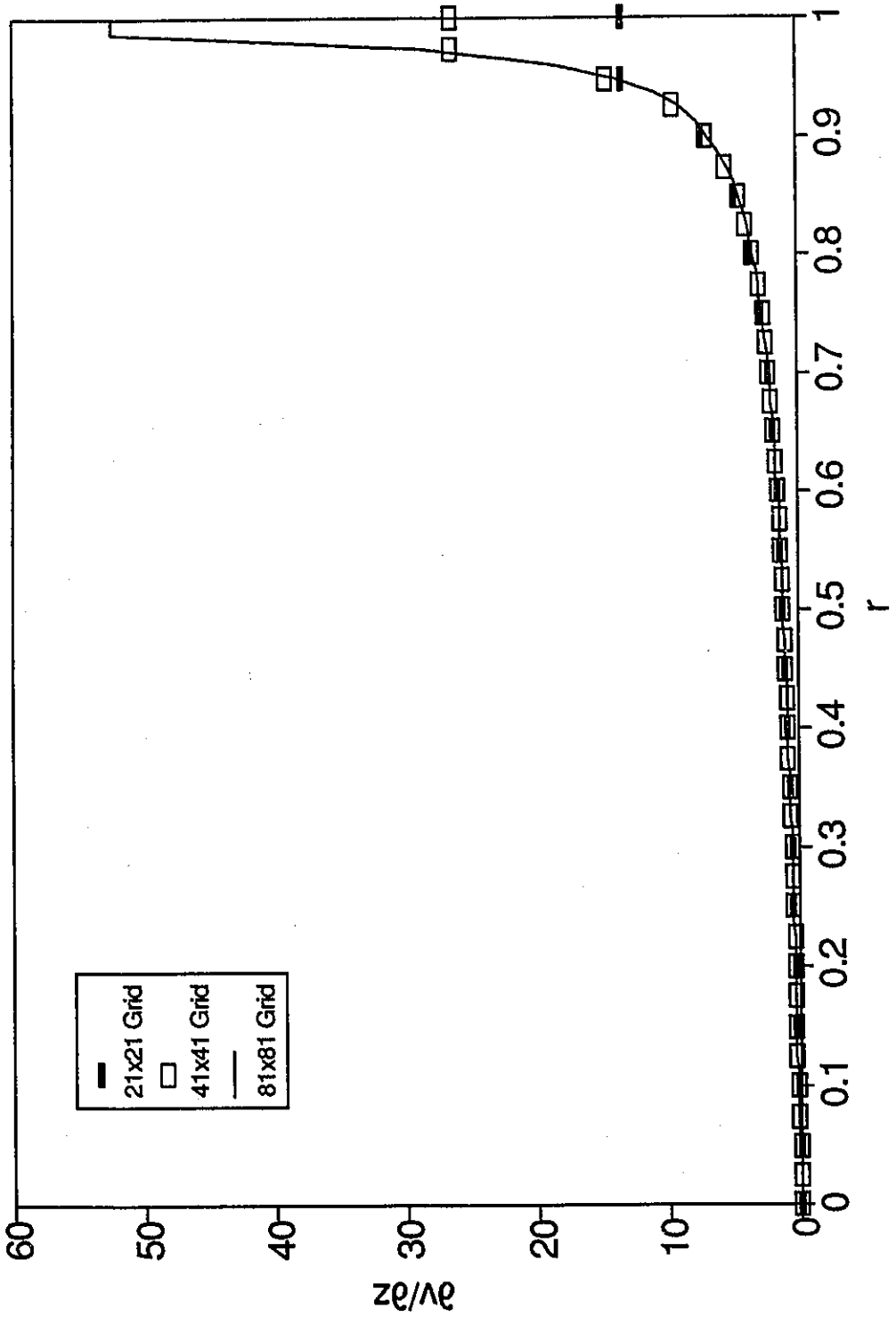

Figure 4.31 Gradient of the θ-velocity on the Disk Calculated Using Pao's Method for Re = 1

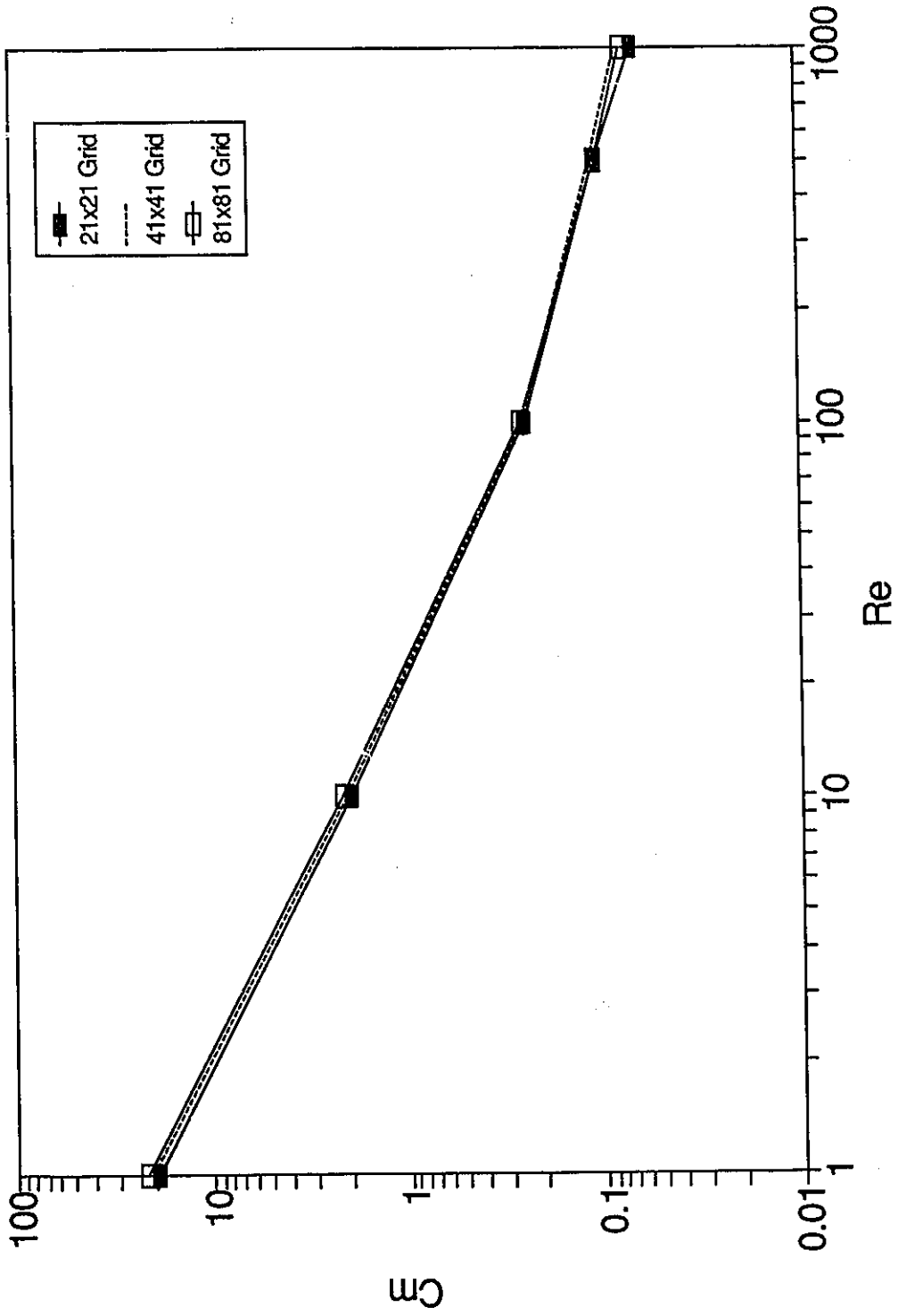

Figure 4.32 Torque Coefficient Calculated Using the Symmetry Boundary Conditions

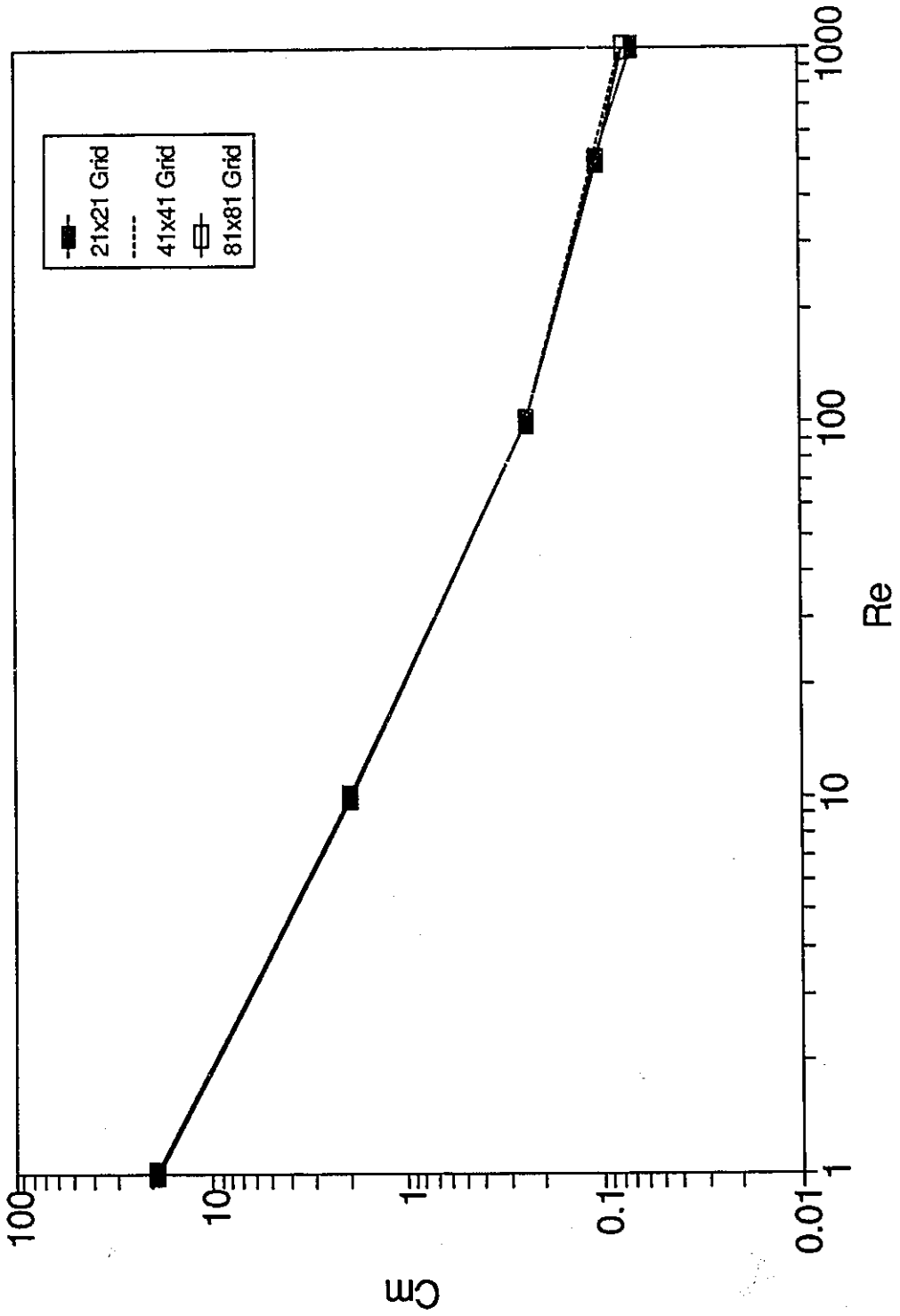

Figure 4.33 Torque Coefficient Calculated Using the Cylindrical Couette Flow Boundary Conditions

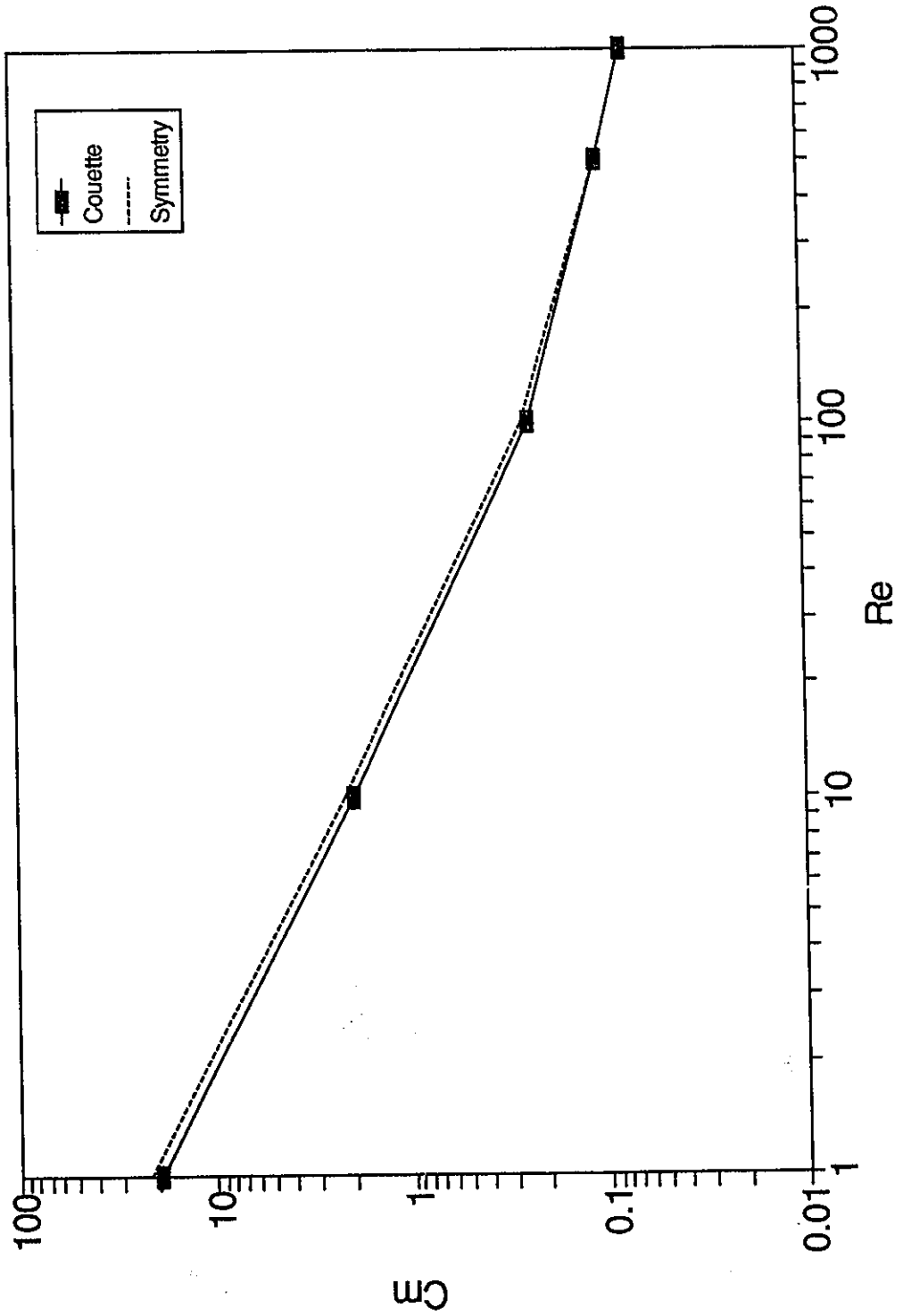

Figure 4.34 Comparison of Torque Coefficient Calculated by Couette Flow and Symmetry Boundary Conditions on an 81x81 Grid

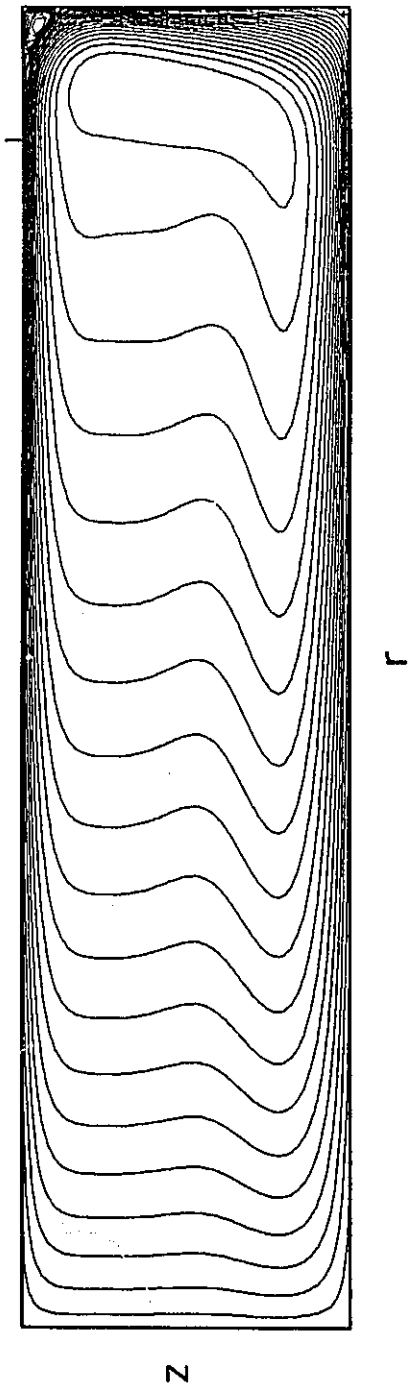

Figure 4.35 Stream Function Contours for Re = 10 000 and δ = 0.25 Calculated Using the Couette Flow Boundary Conditions

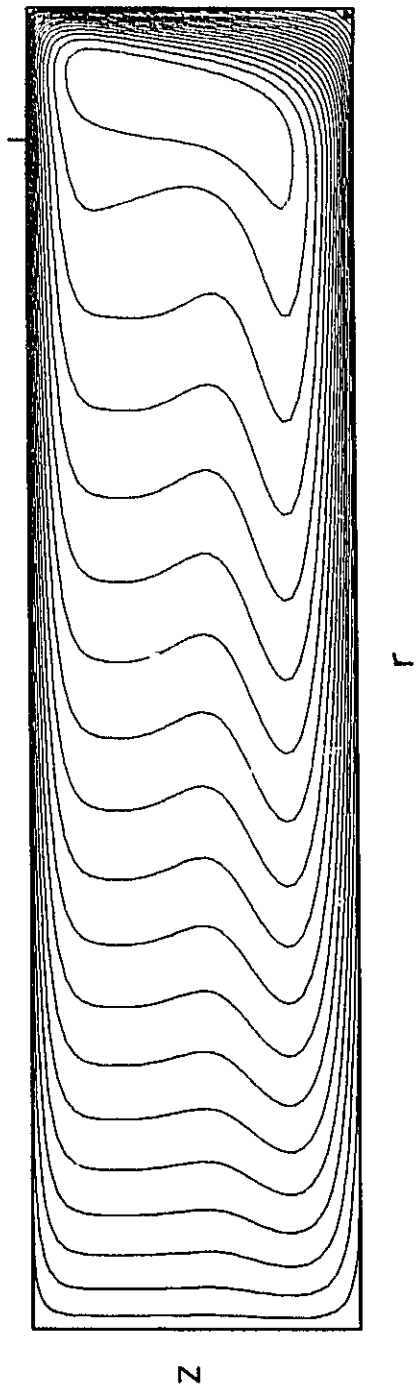

Figure 4.36 Stream Function Contours for Re = 10 000 and δ = 0.25 Calculated Using the Symmetry Boundary Conditions

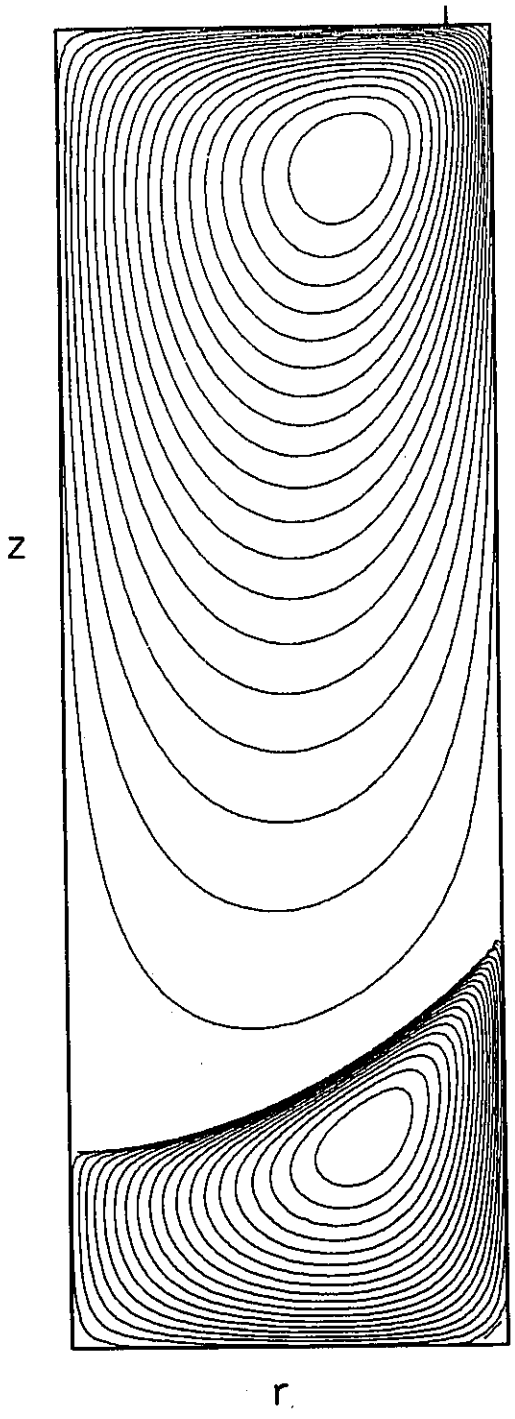

Figure 4.37 Stream Function Contours for Re = 100 and δ = 3.00

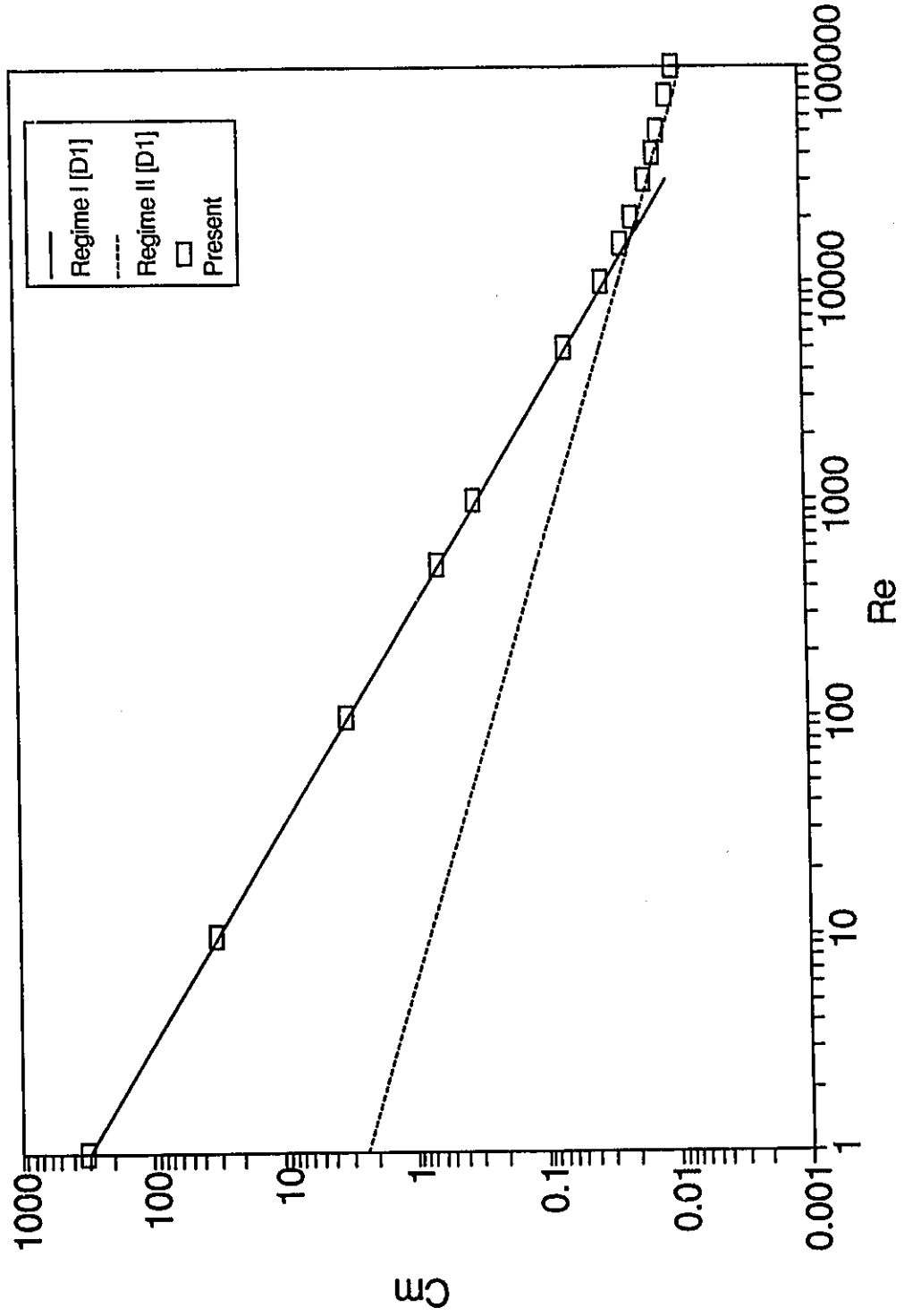

Figure 4.38 Comparison of Computational with Experimental Results of Daily and Nece [D1], Equations (4.18) and (4.19), for $\delta = 0.02$

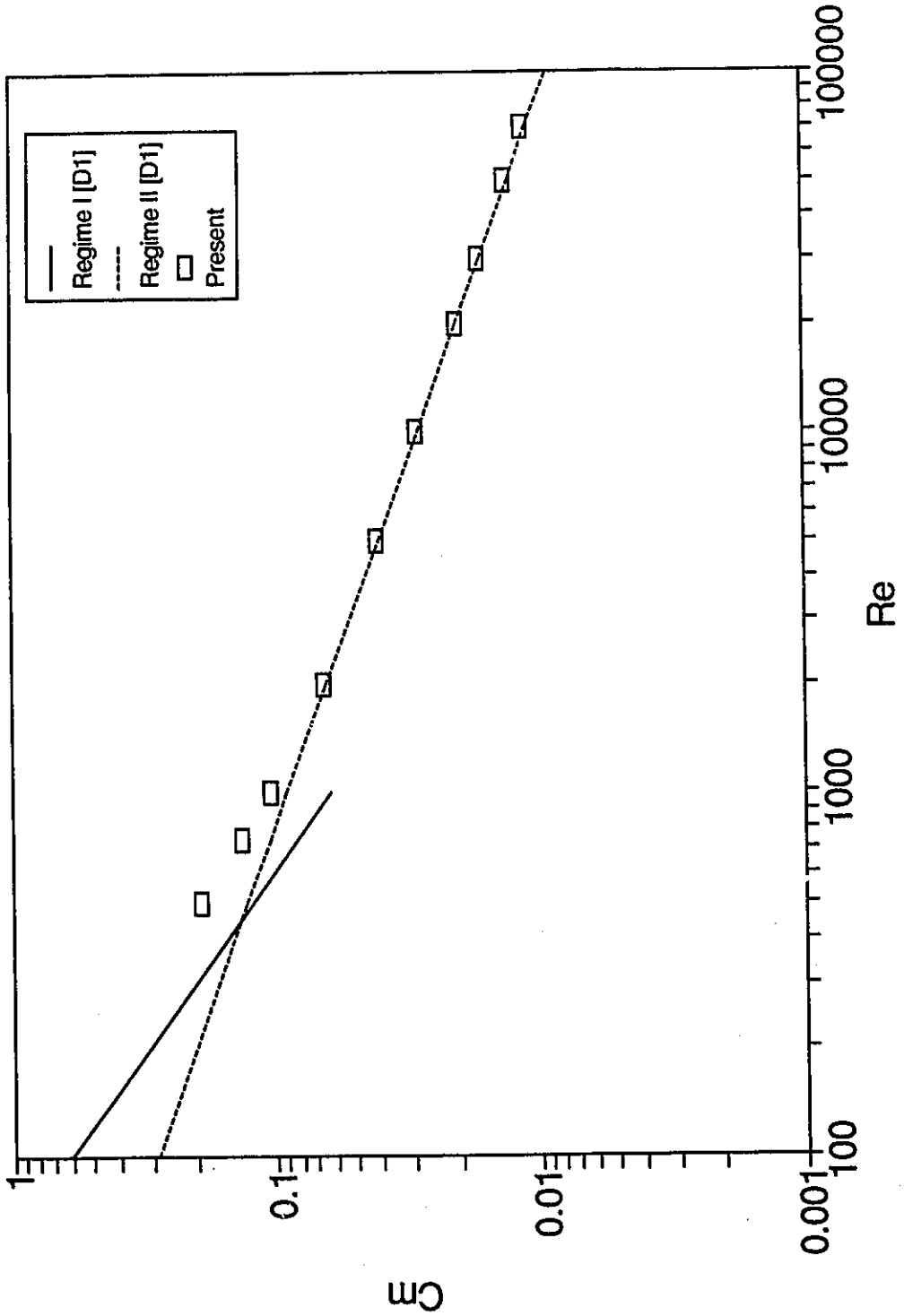

Figure 4.39 Comparison of Computational with Experimental Results of Daily and Nece [D1], Equations (4.18) and (4.19), for $\delta = 0.10$

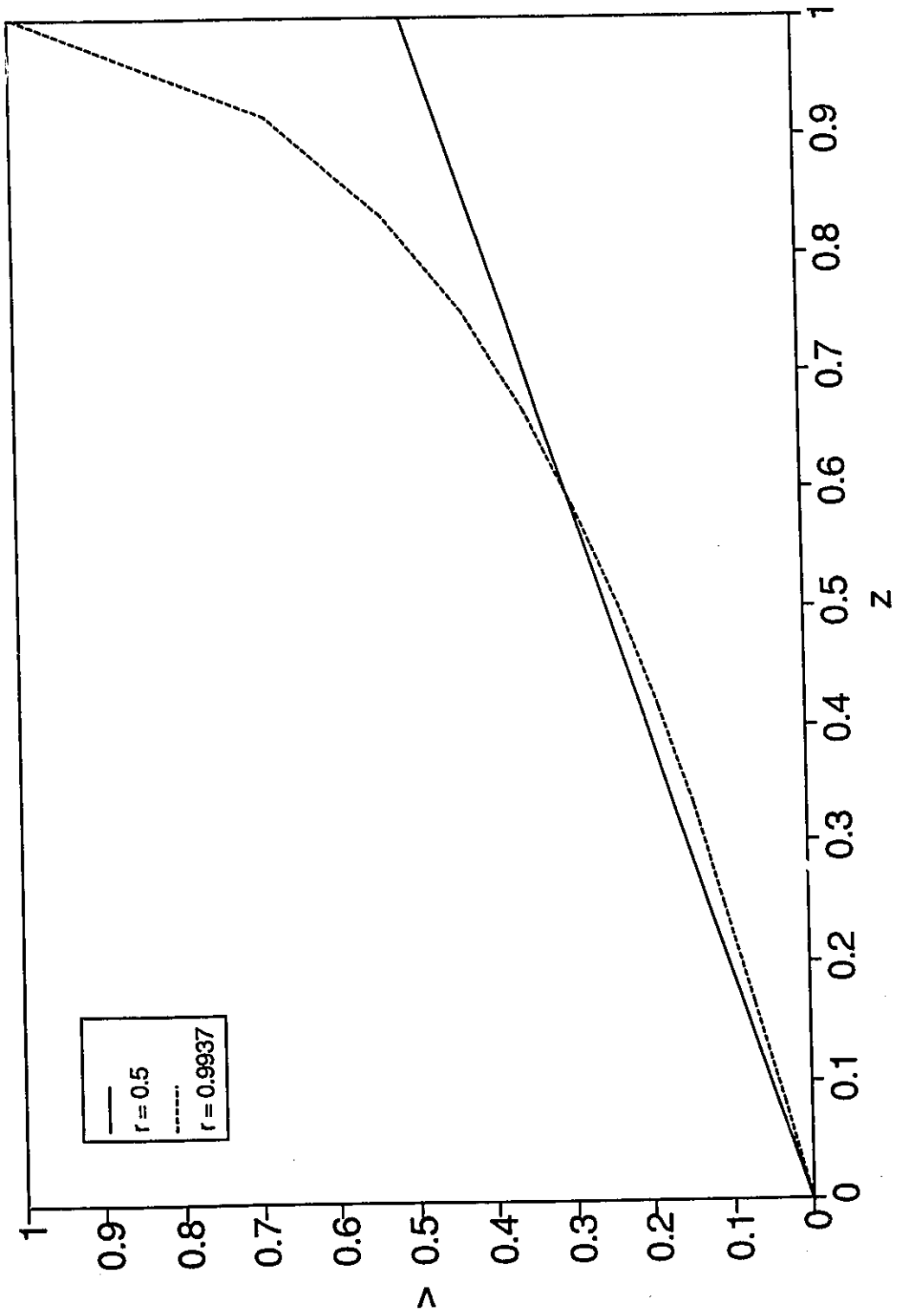

Figure 4.40 Profile of the v-component of Velocity for Re = 100 and δ = 0.02

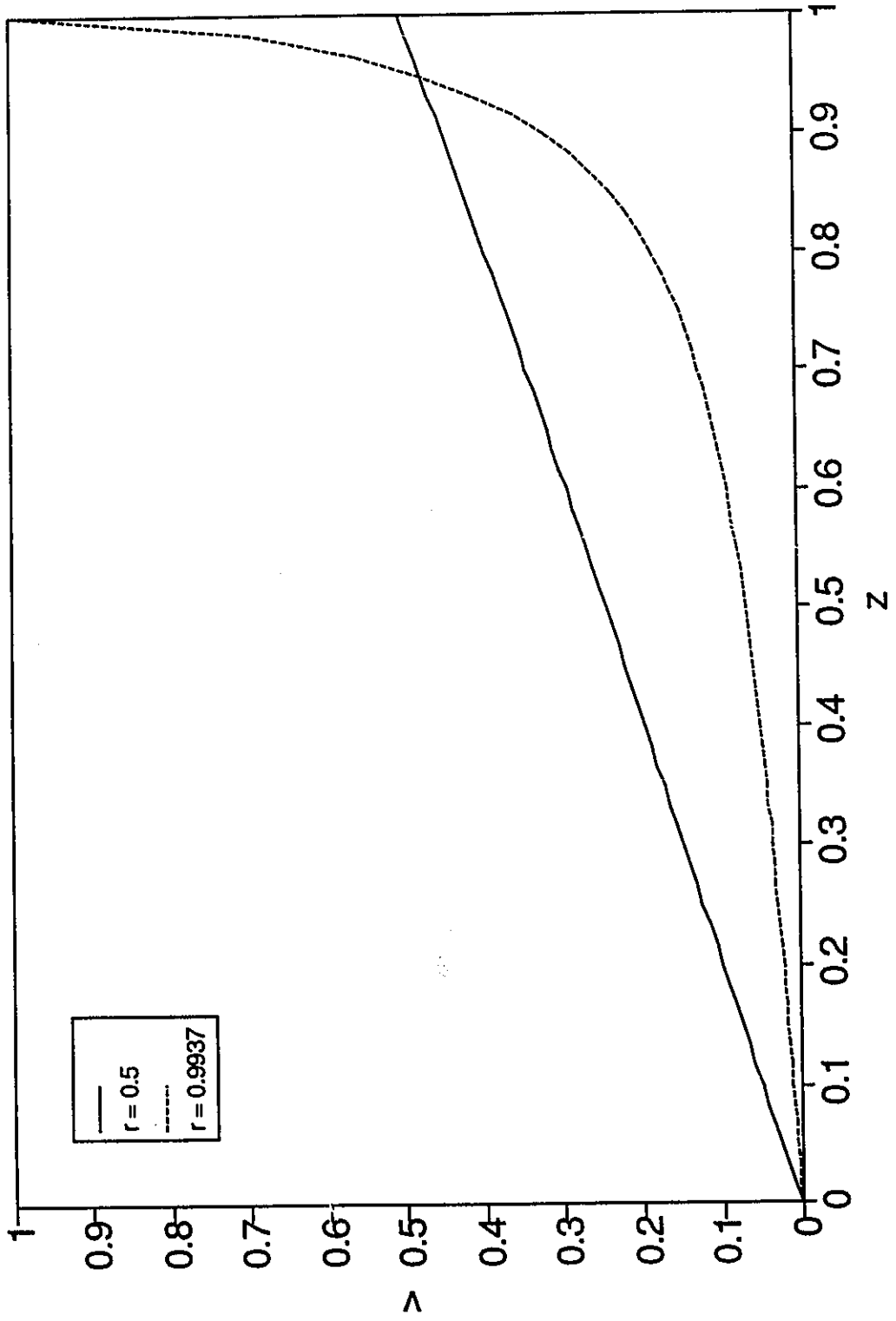

Figure 4.41 Profile of v-component of Velocity for Re = 500 and $\delta = 0.10$

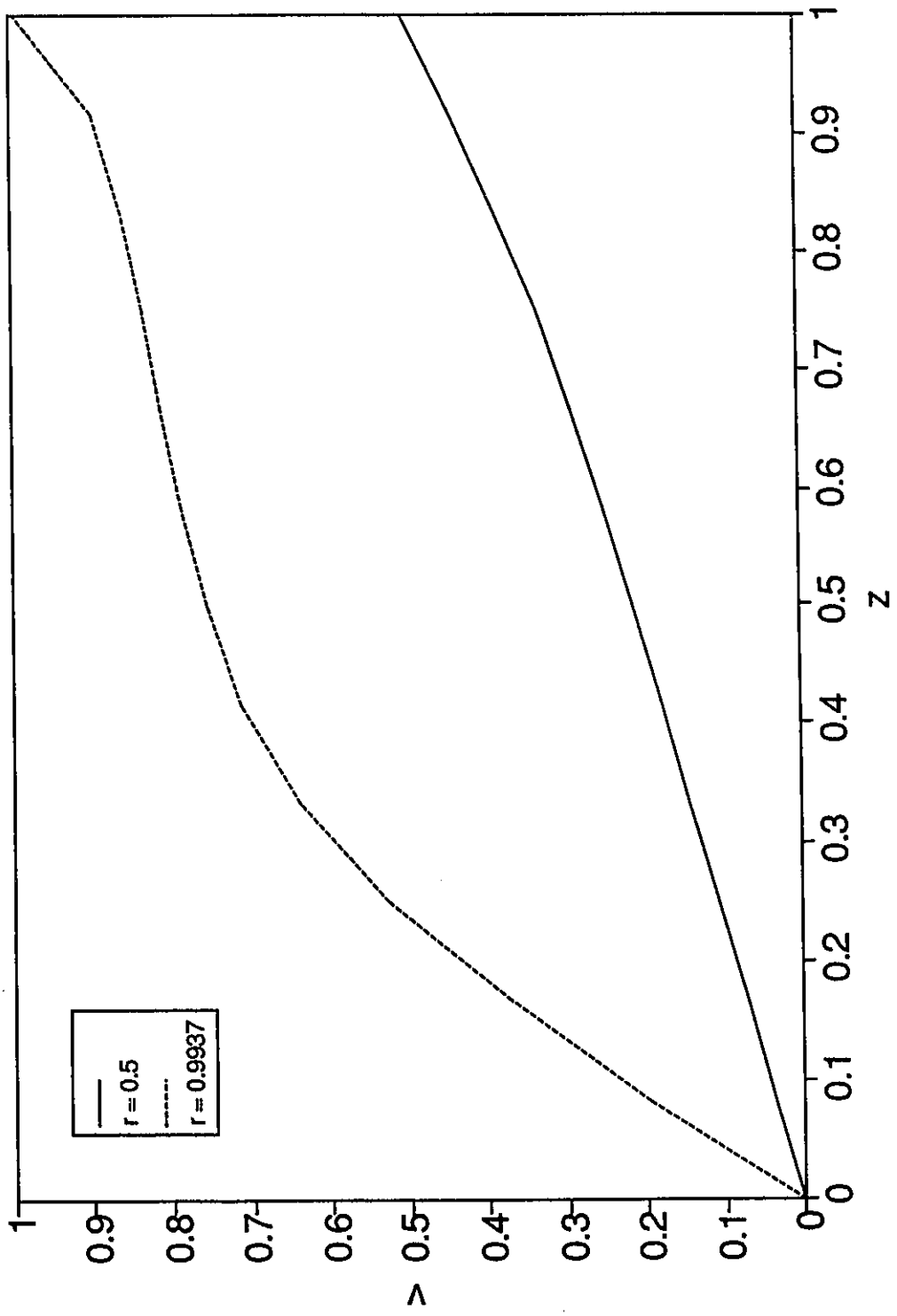

Figure 4.42 Profile of the v-component of Velocity for Re = 30 000 and $\delta = 0.02$

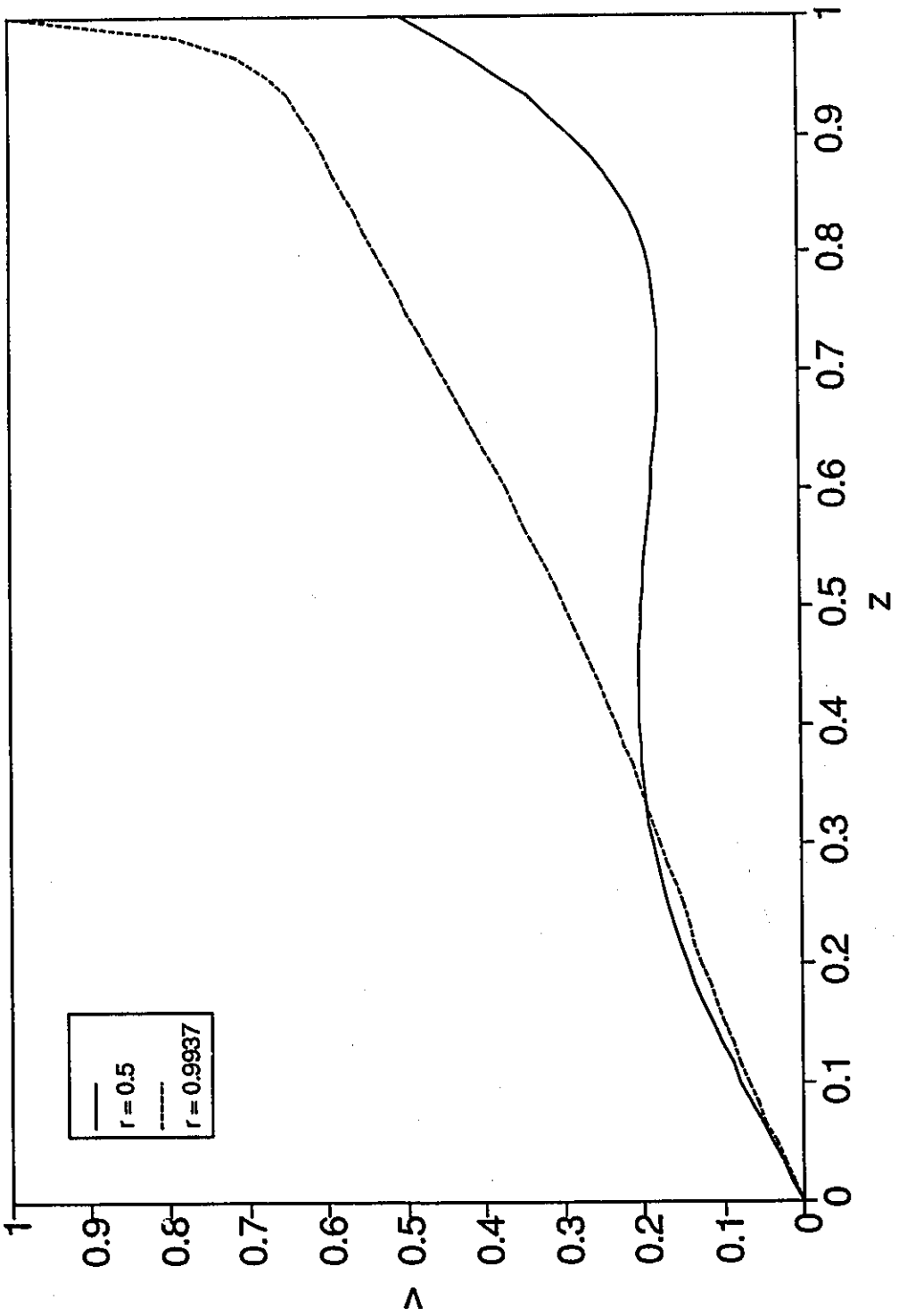

Figure 4.43 Profile of the v-component of Velocity for Re = 10 000 and $\delta = 0.10$

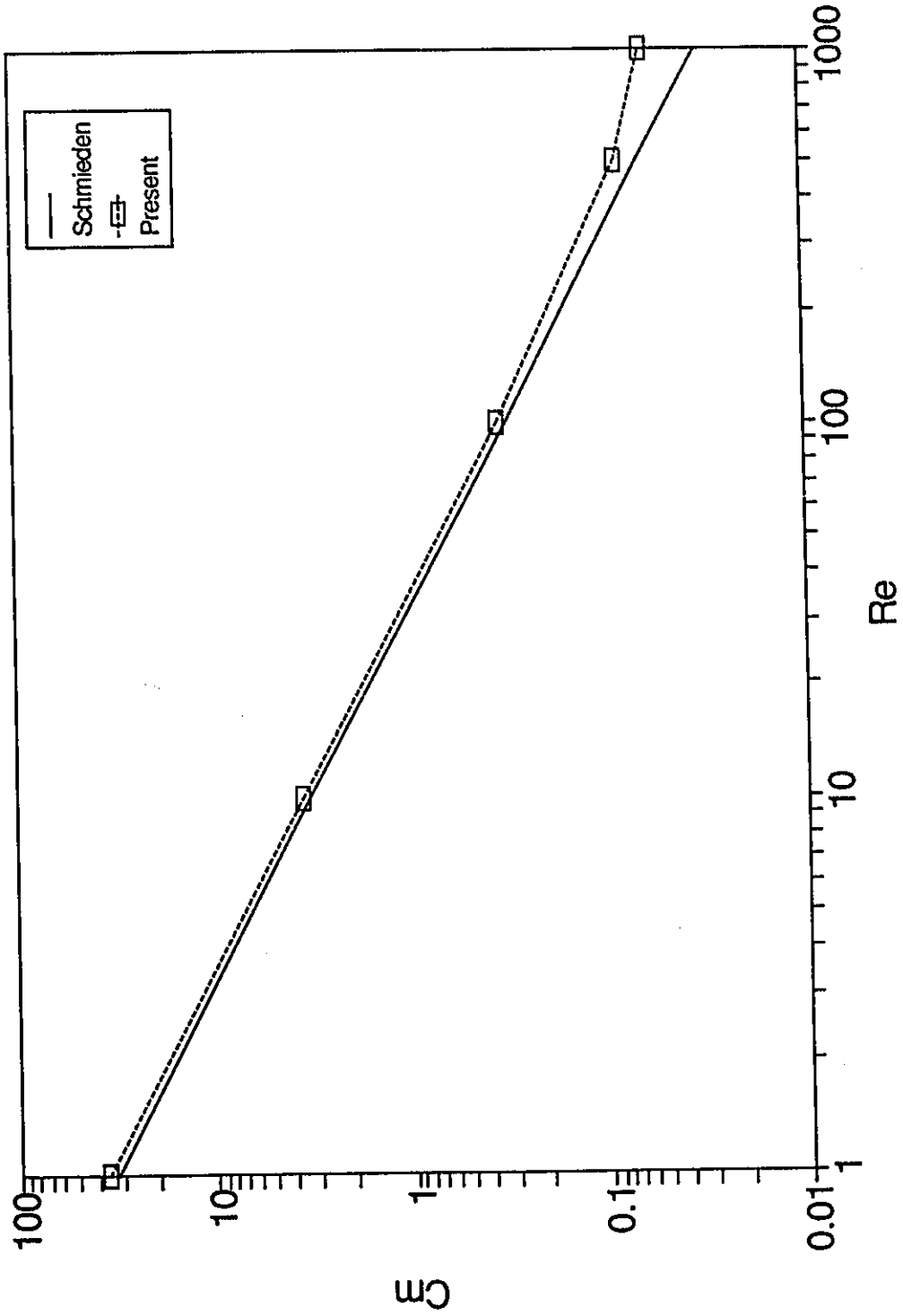

Figure 4.44 Comparison of Torque Coefficients Calculated by Schmieden [S2] and Computational Method for $\delta = 0.15$ and $b = 0.10$

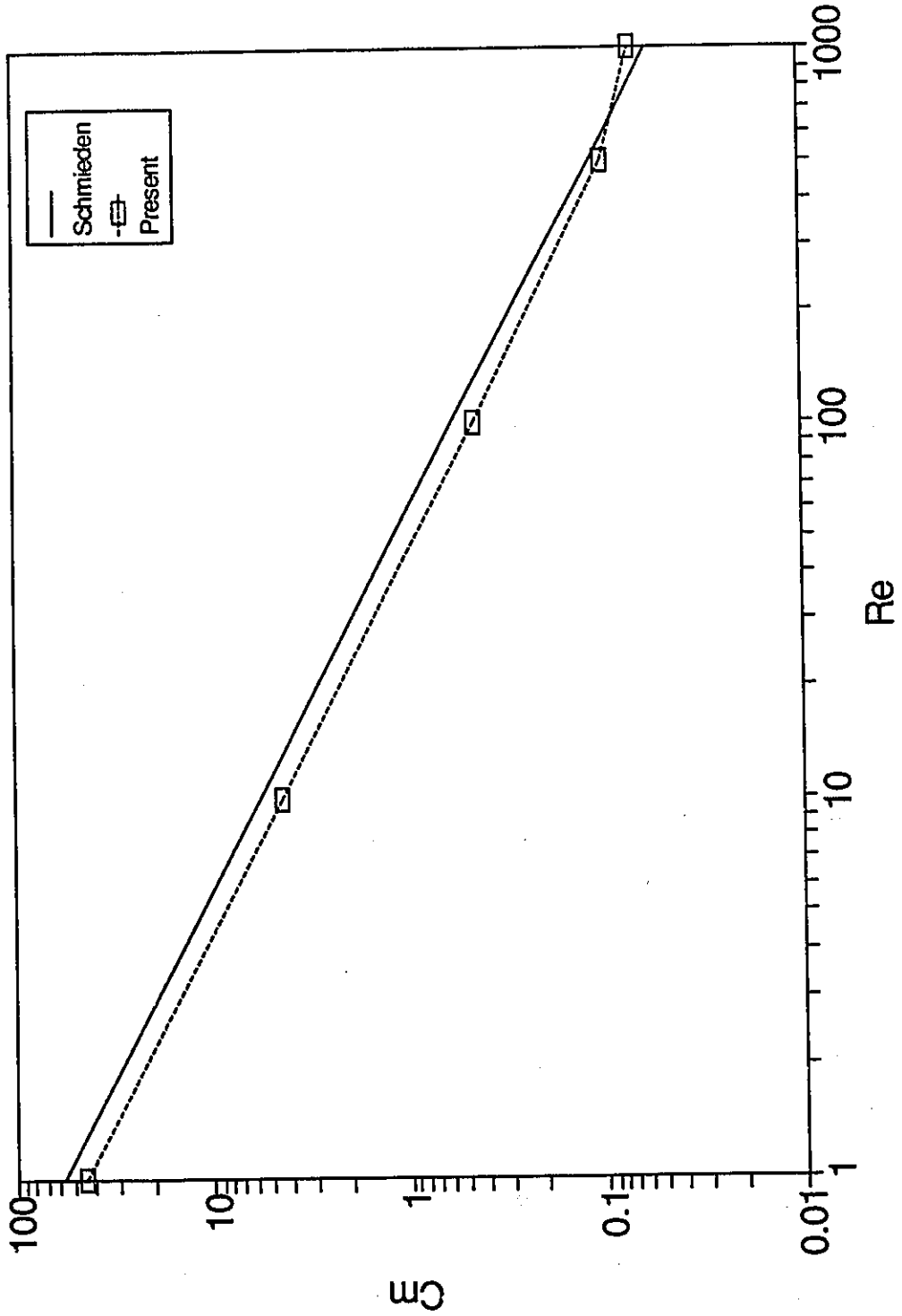

Figure 4.45 Comparison of Torque Coefficients Calculated by Schmieden [S2] and Computational Method for $\delta = 0.15$ and $b = 0.06$

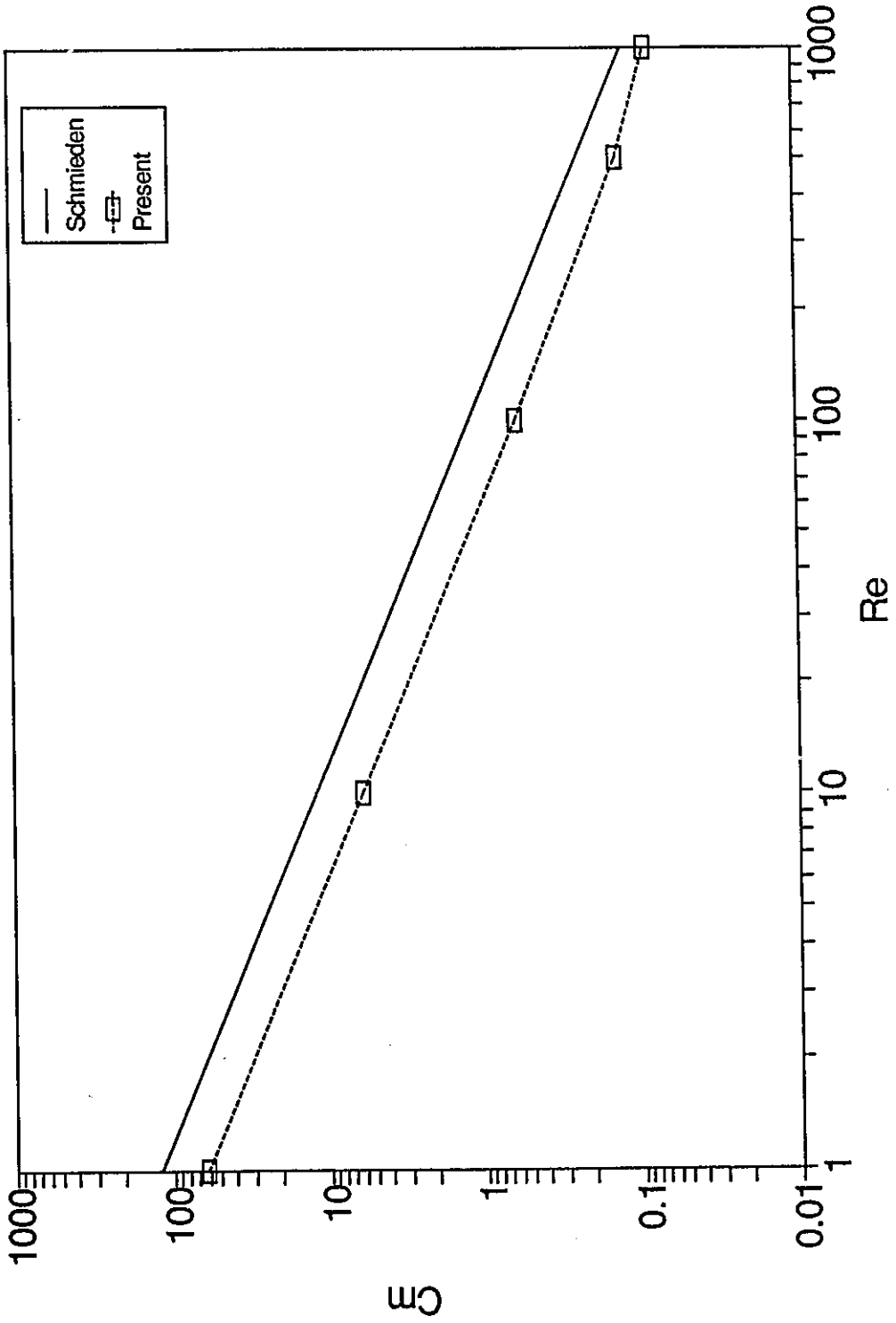

Figure 4.46 Comparison of Torque Coefficients Calculated by Schmieden [S2] and Computational Method for $\delta = 0.15$ and $b = 0.02$

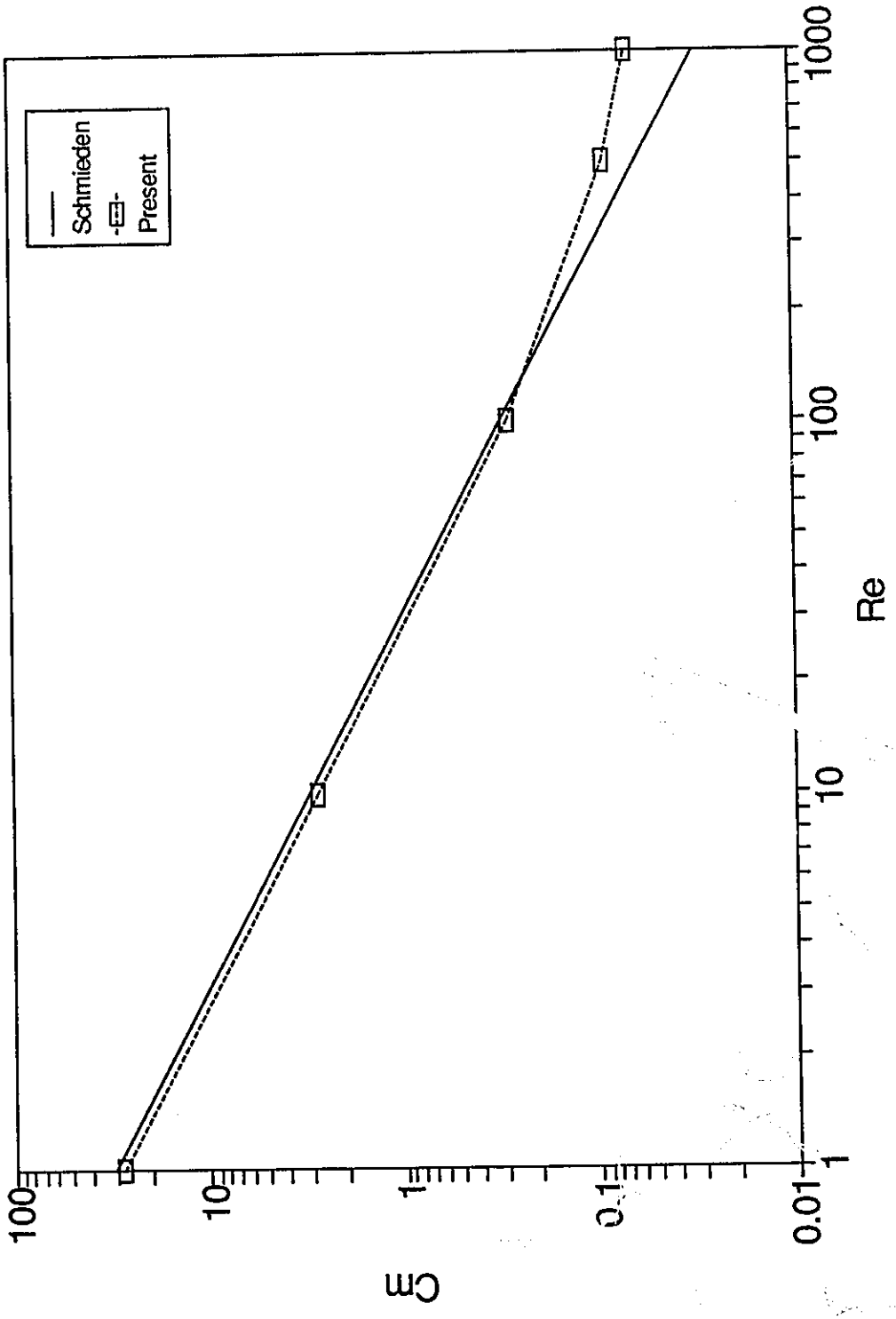

Figure 4.47 Comparison of Torque Coefficients Calculated by Schmieden [S2] and Computational Method for $\delta = 0.25$ and $b = 0.10$

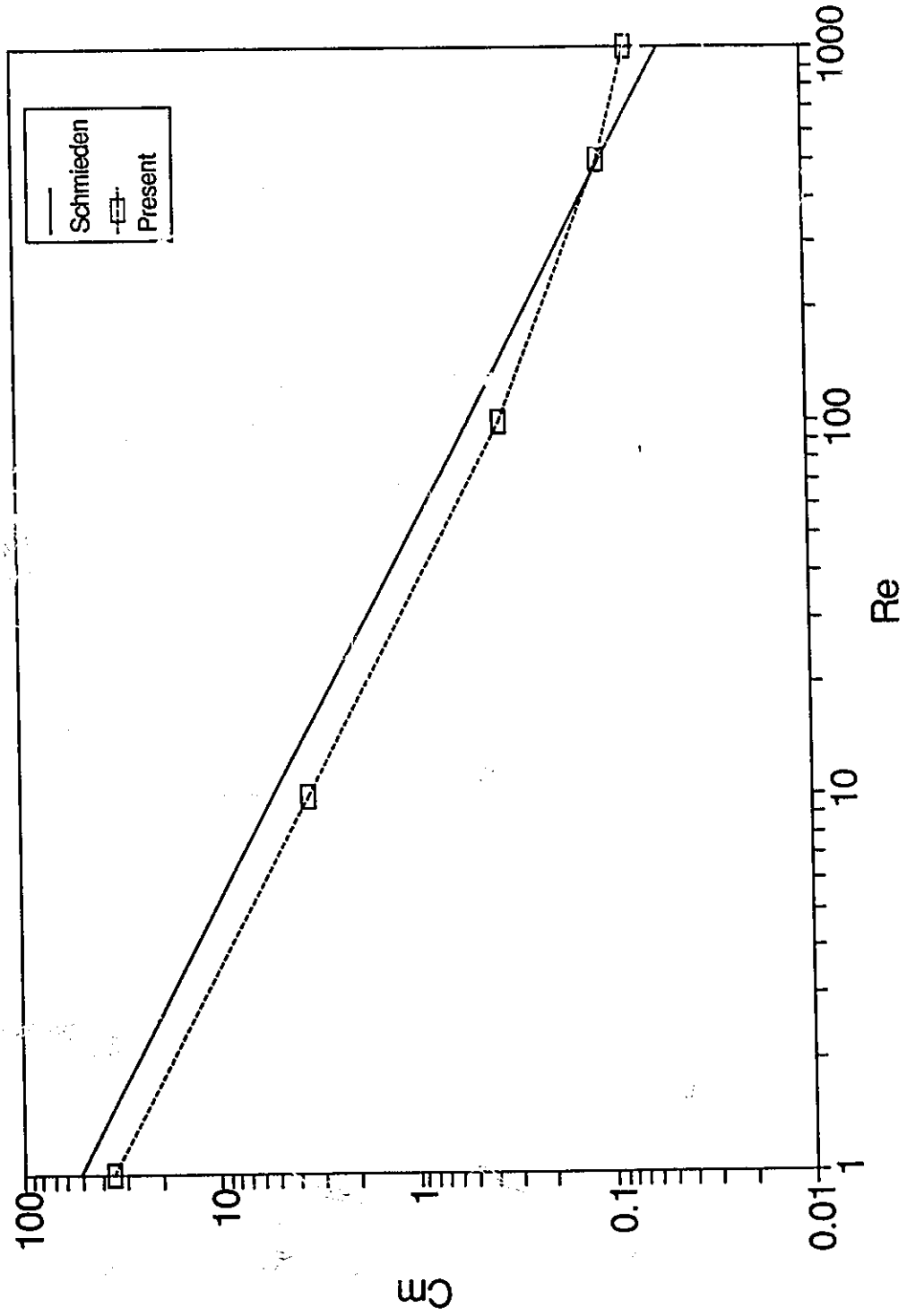

Figure 4.48 Comparison of Torque Coefficients Calculated by Schmieden [S2] and Computational Method for $\delta = 0.25$ and $b = 0.06$

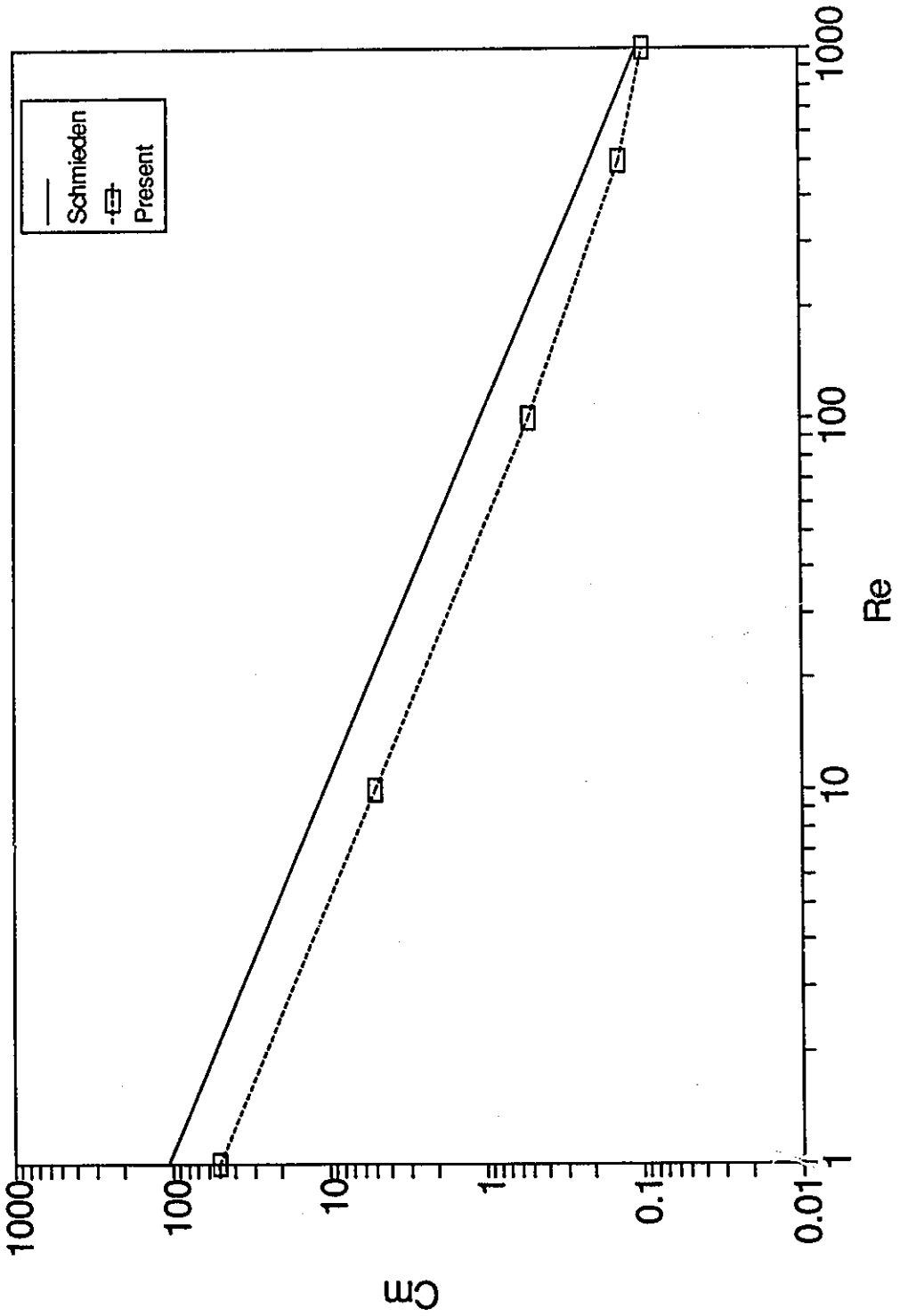

Figure 4.49 Comparison of Torque Coefficients Calculated by Schmieden [S2] and Computational Method for $\delta = 0.25$ and $b = 0.02$

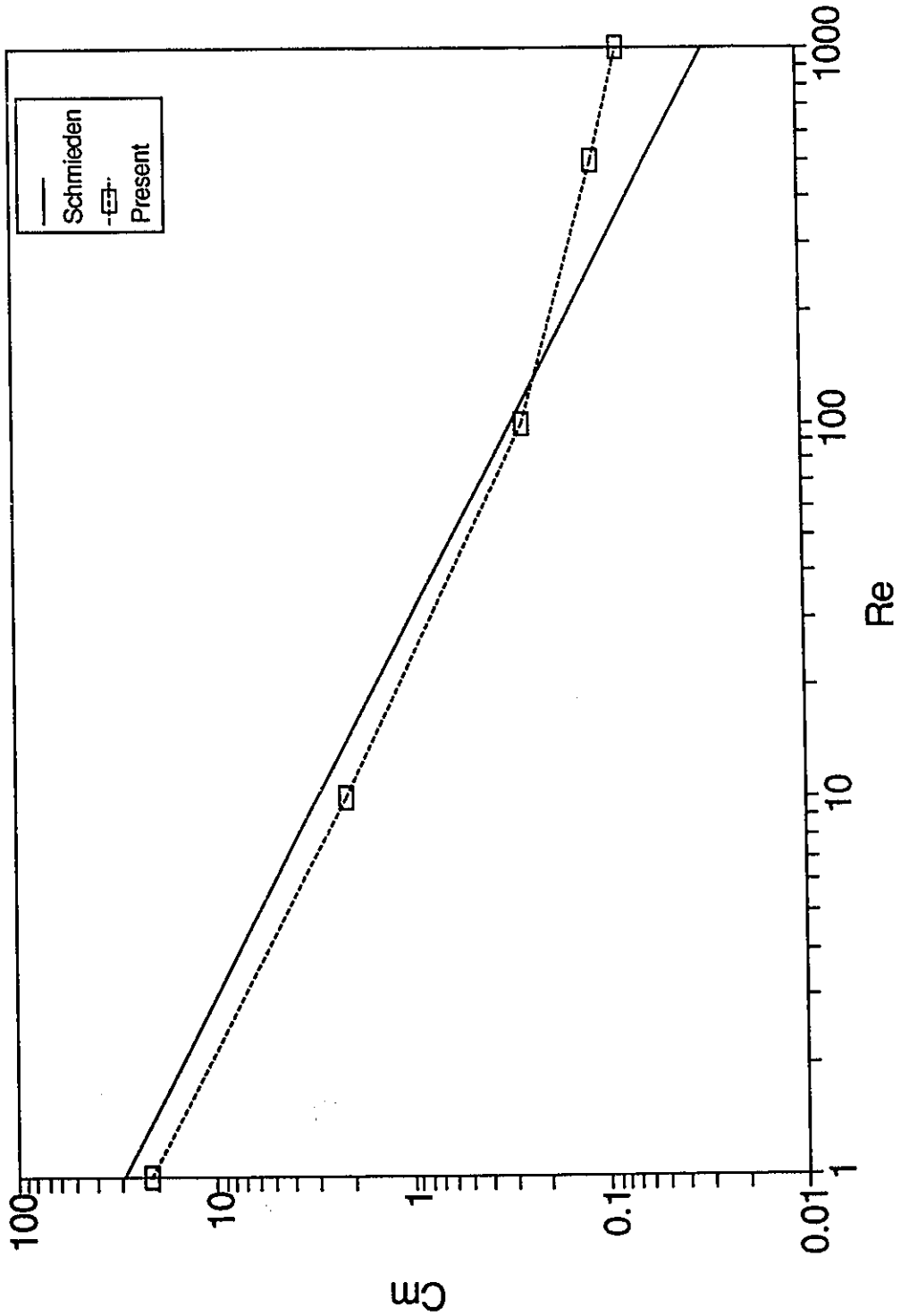

Figure 4.50 Comparison of Torque Coefficients Calculated by Schmieden [S2] and Computational Method for $\delta = 2.00$ and $b = 0.10$

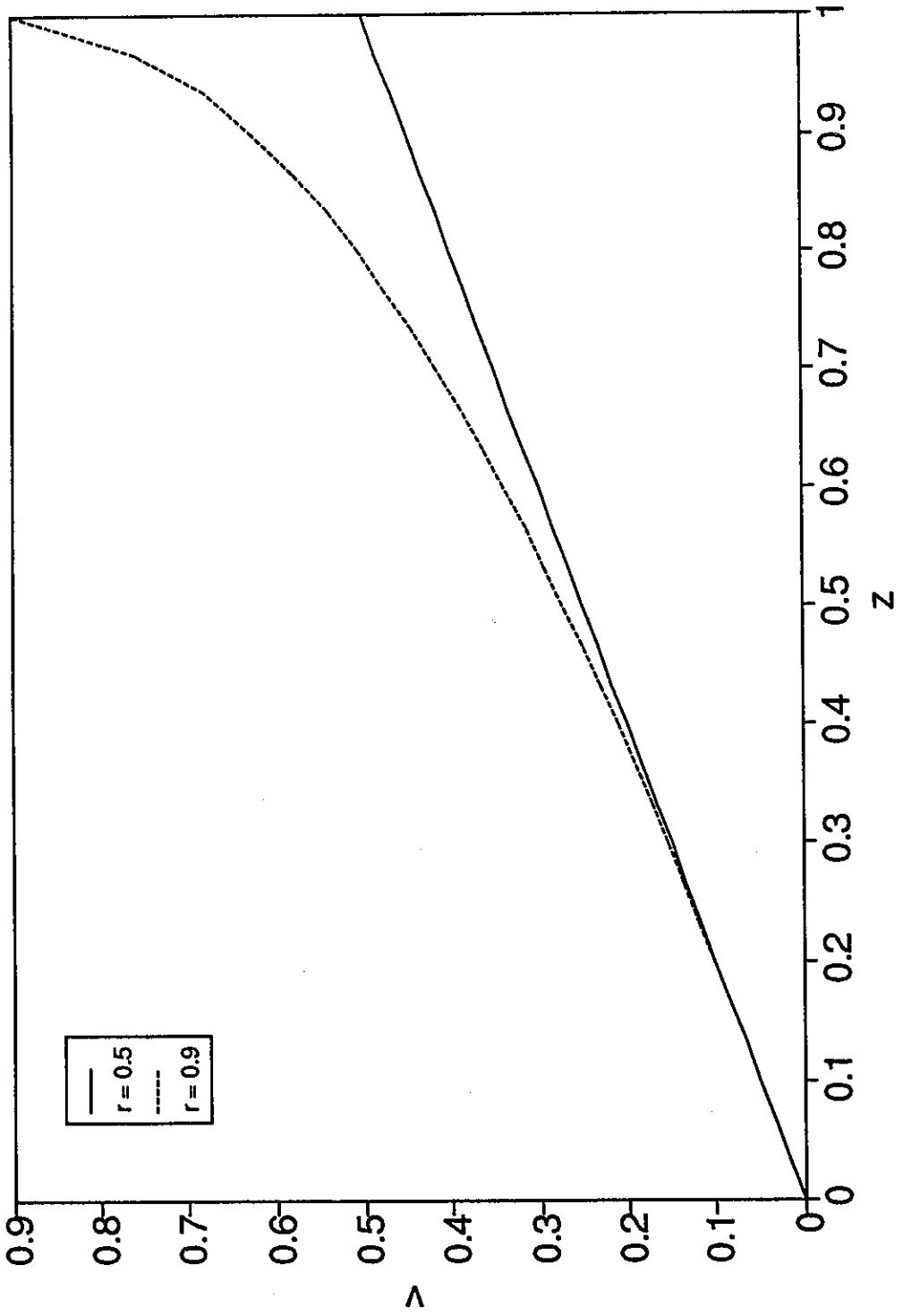

Figure 4.51 Profile of the v-component of Velocity for Re = 1, δ = 0.15 and b = 0.10

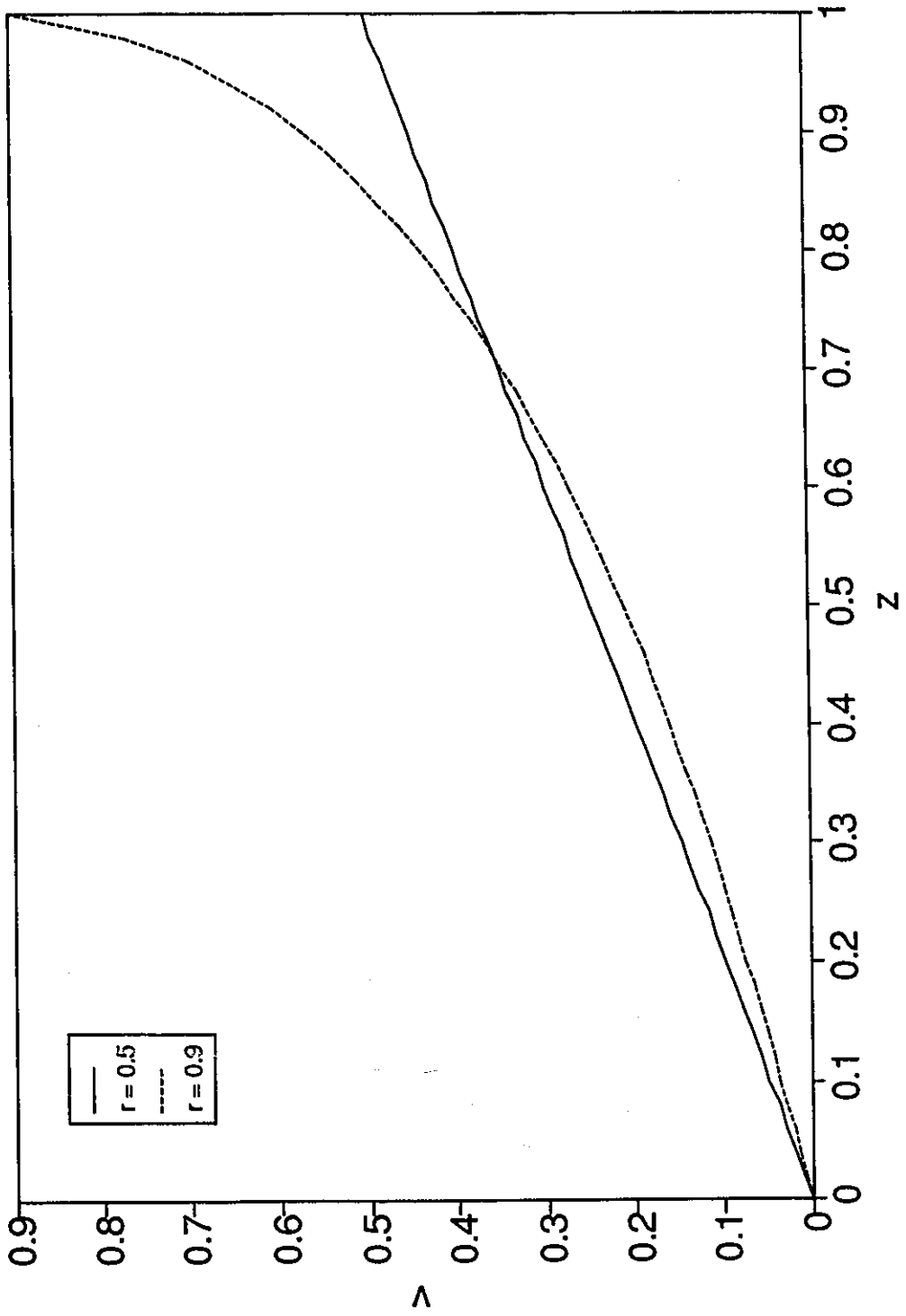

Figure 4.52 Profile of the v-component of Velocity for Re = 1, δ = 0.25 and b = 0.10

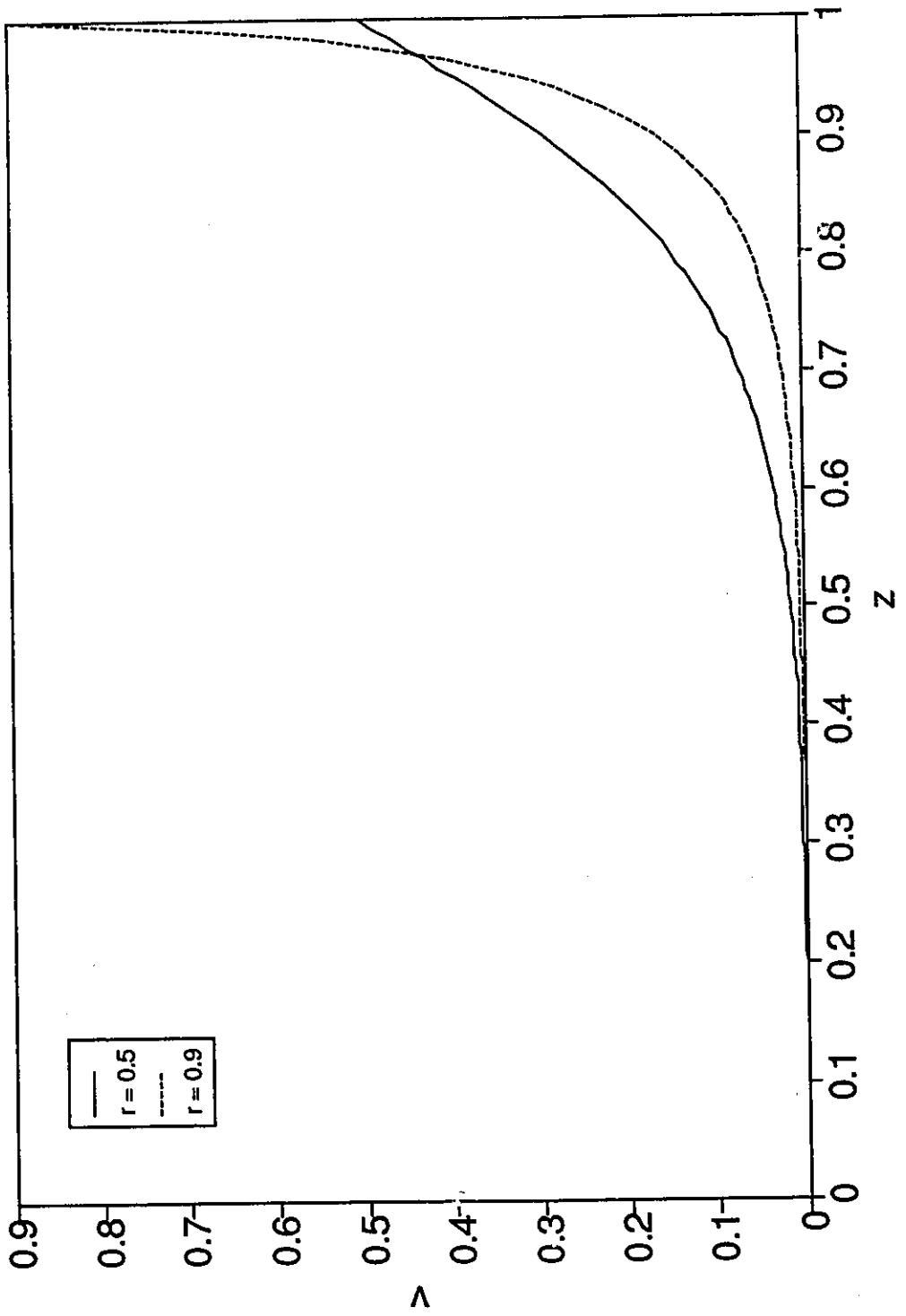

Figure 4.53 Profile of the v-component of Velocity for Re = 1, δ = 2.00 and b = 0.10

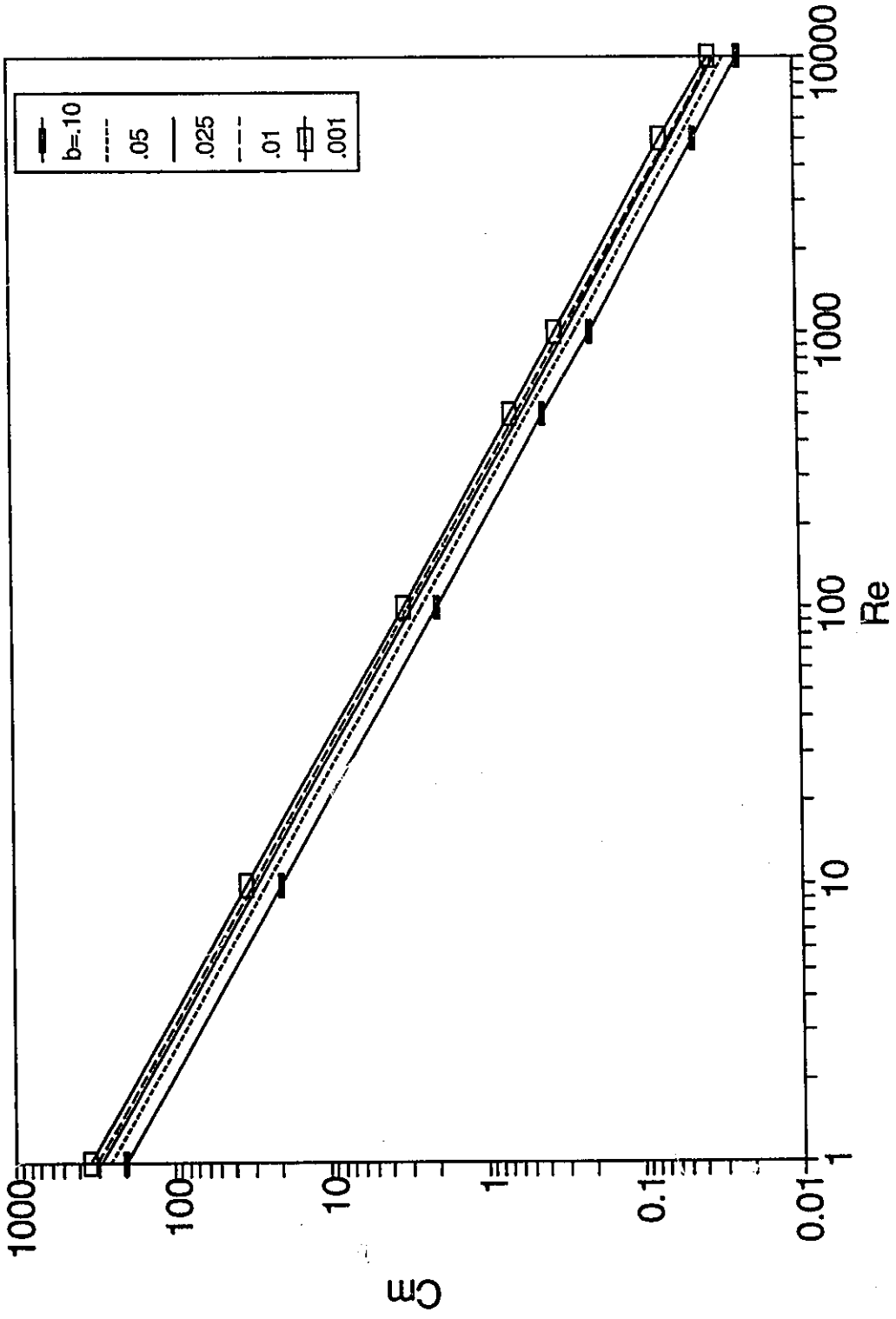

Figure 4.54 Torque Coefficient for $\delta = 0.02$

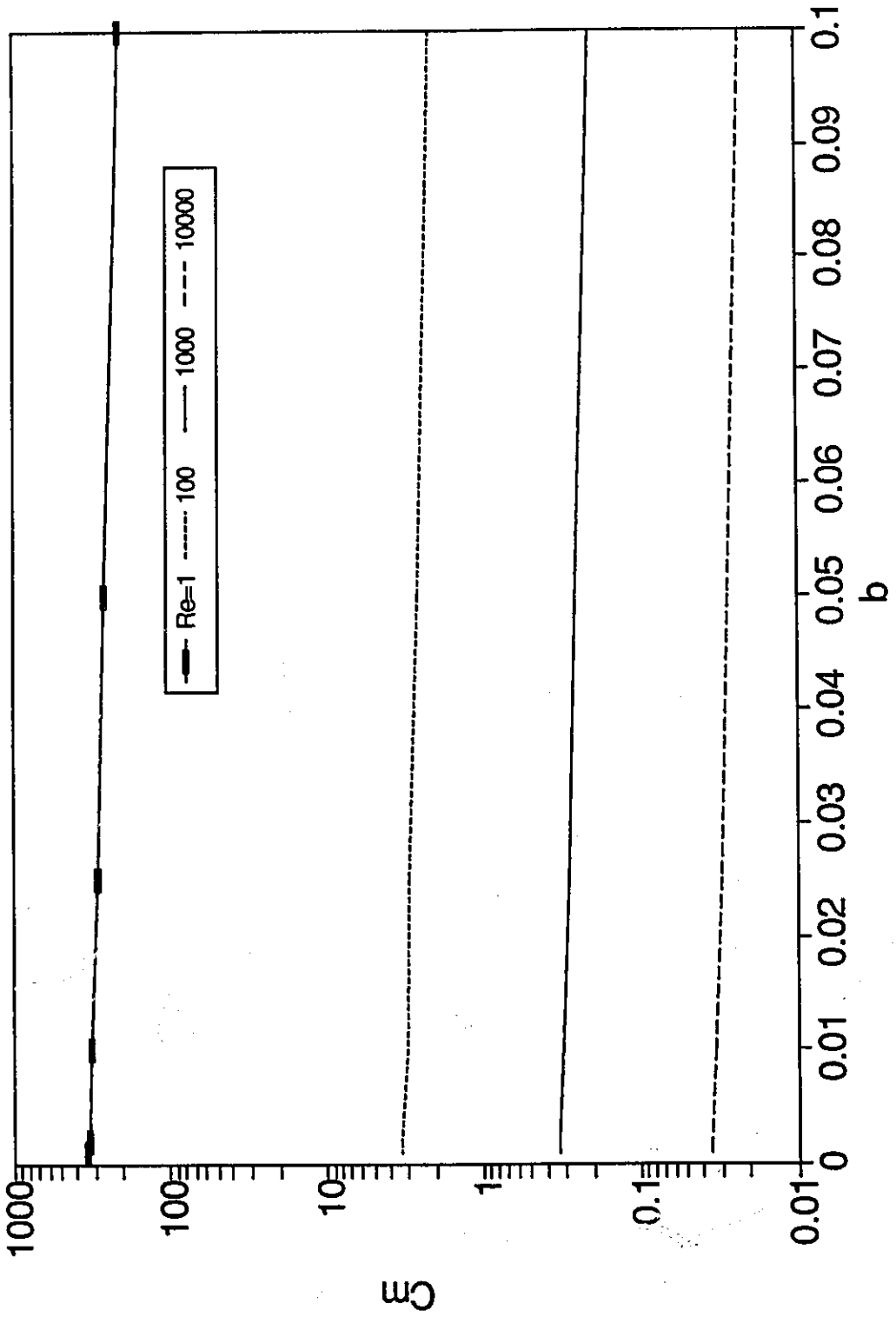

Figure 4.55 Torque Coefficient for $\delta = 0.02$

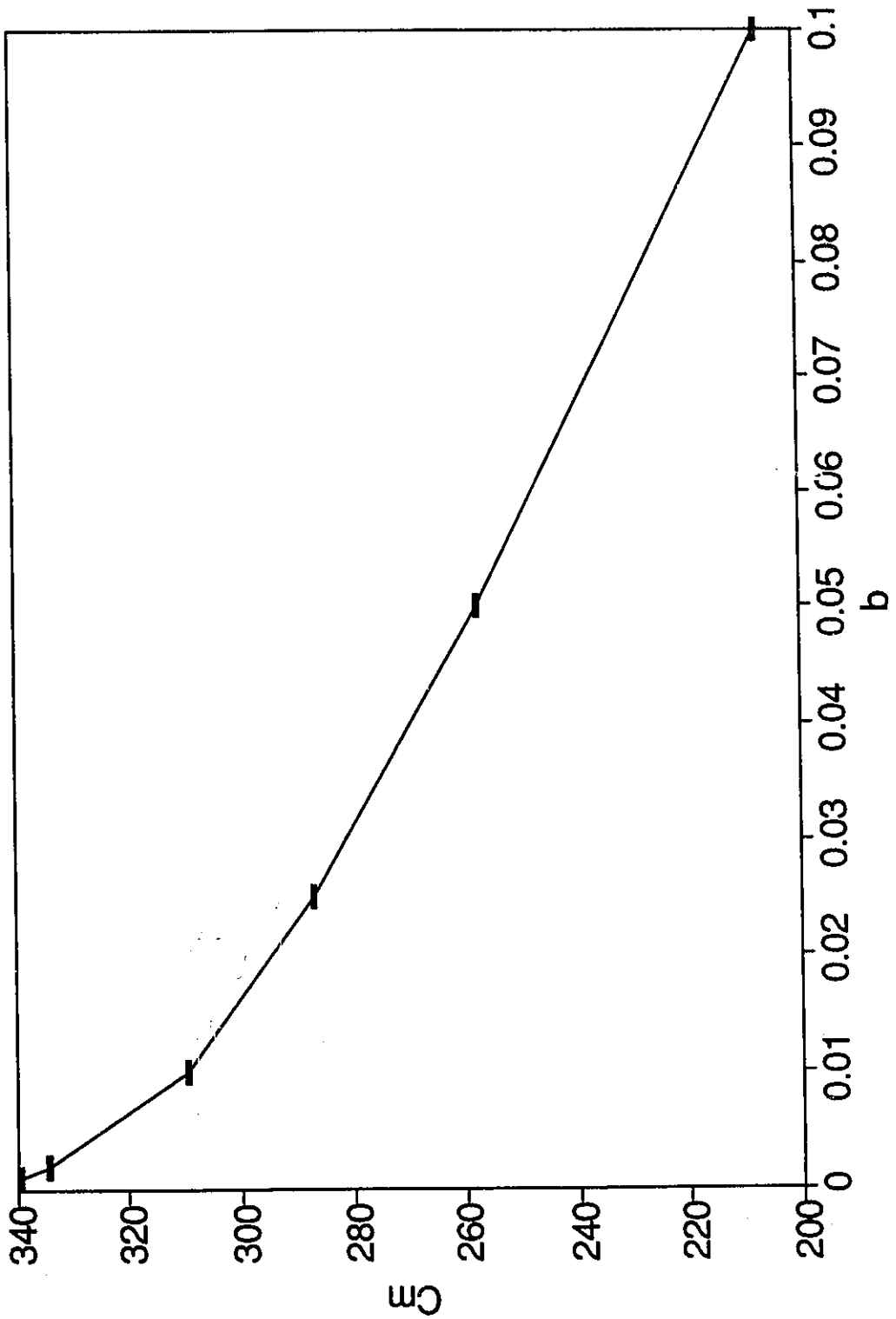

Figure 4.56 Torque Coefficient for $\delta = 0.02$ and Re = 1

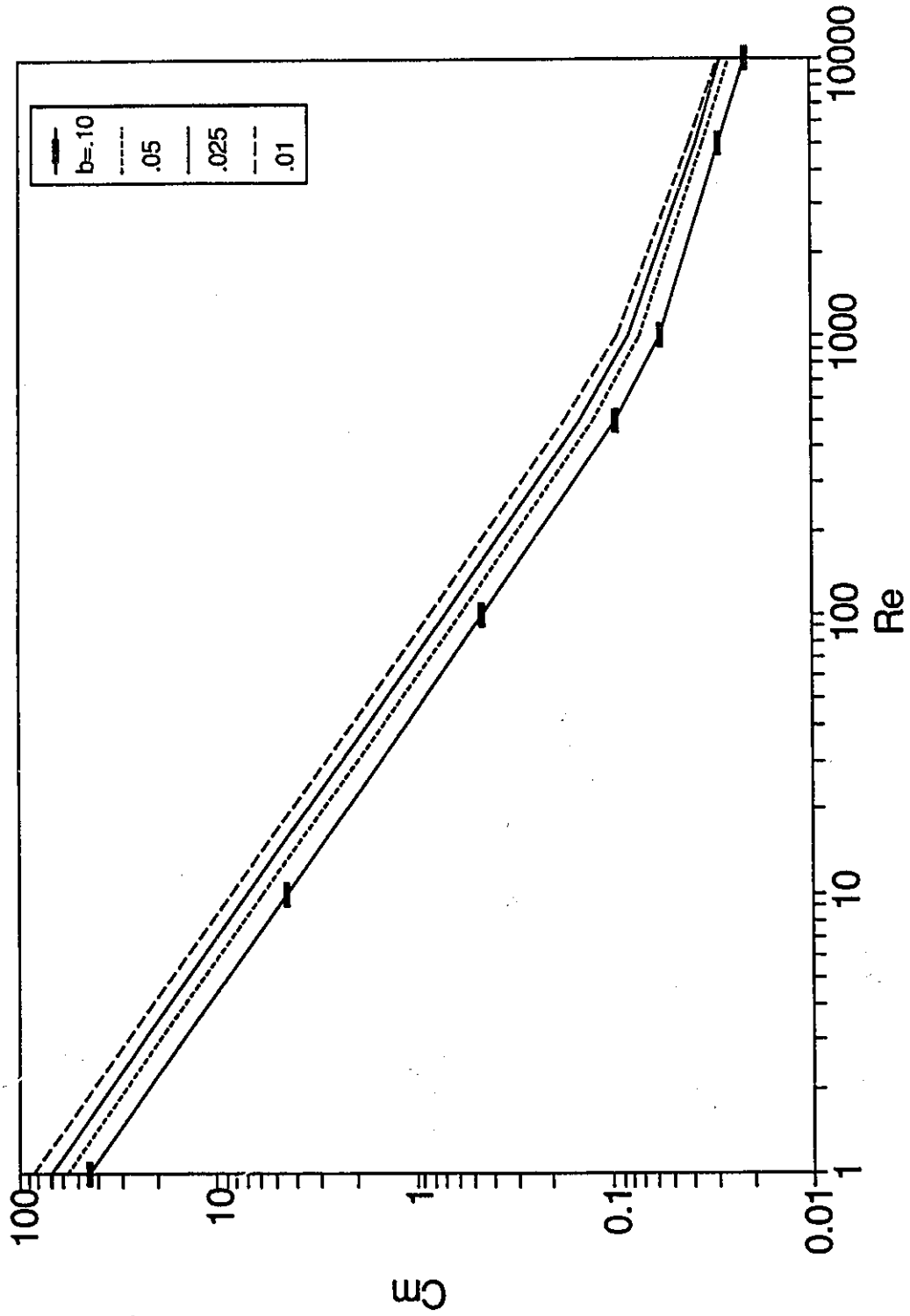

Figure 4.57 Torque Coefficient for $\delta = 0.10$

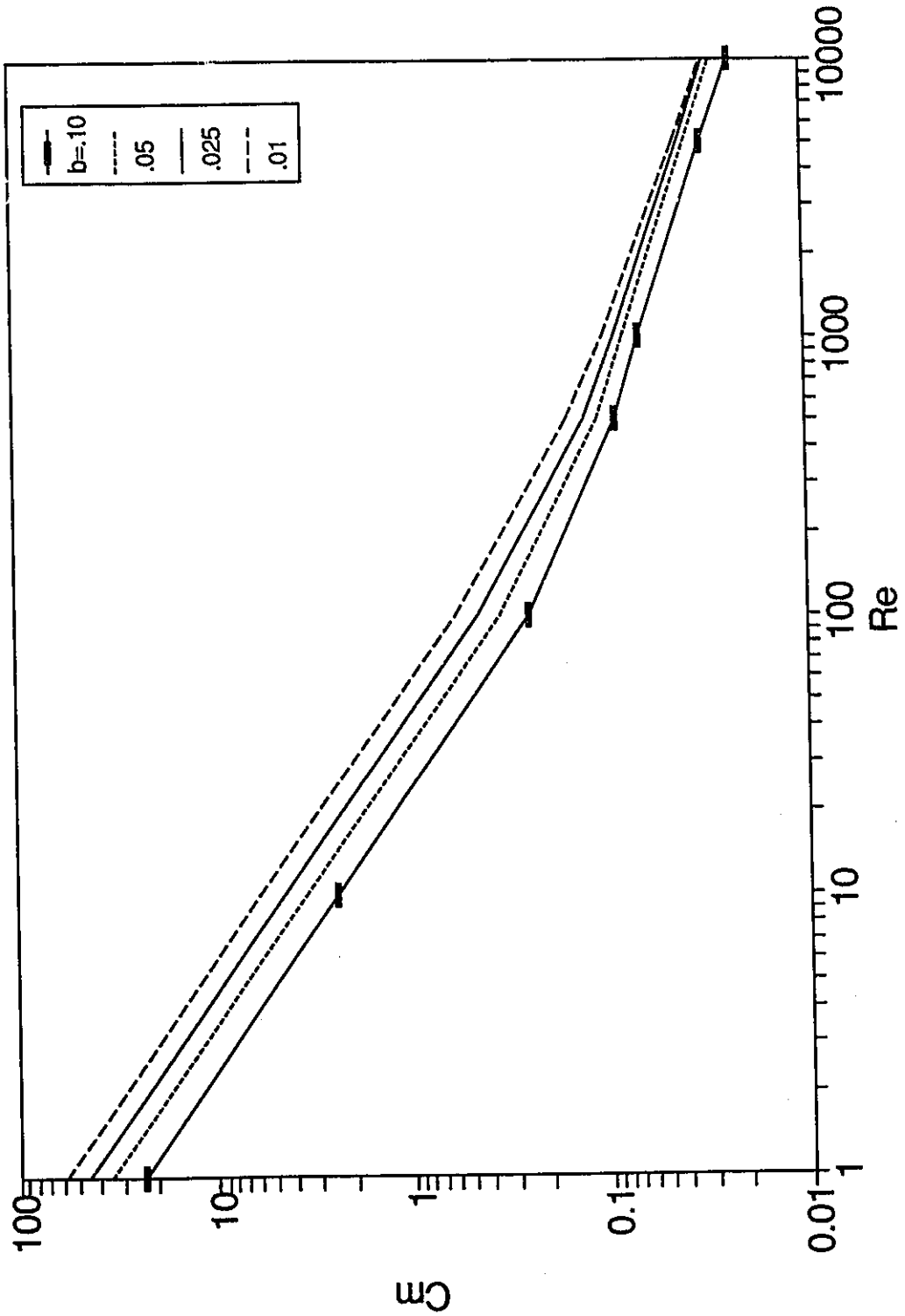

Figure 4.58 Torque Coefficient for $\delta = 0.25$

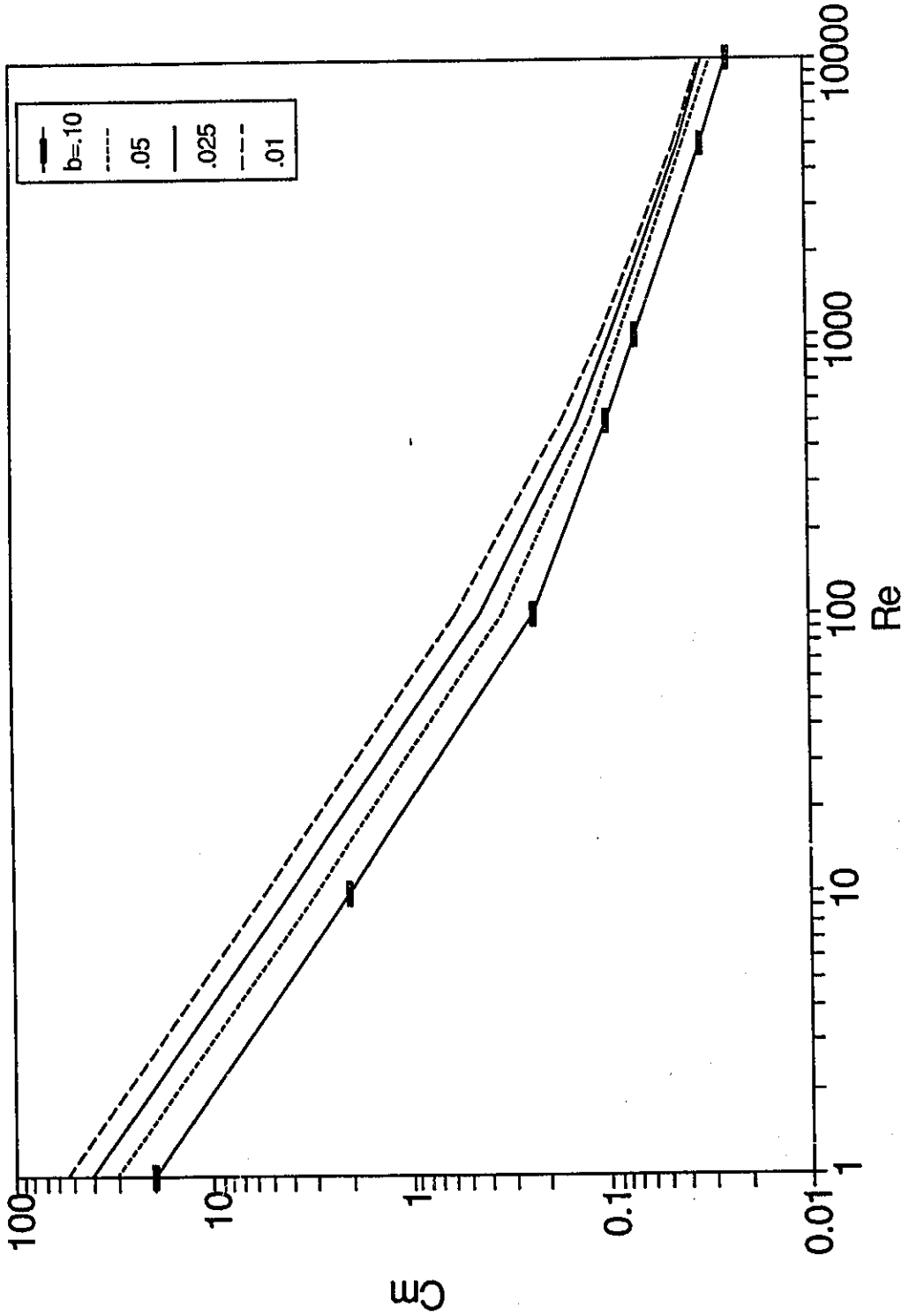

Figure 4.59 Torque Coefficient for $\delta = 0.50$

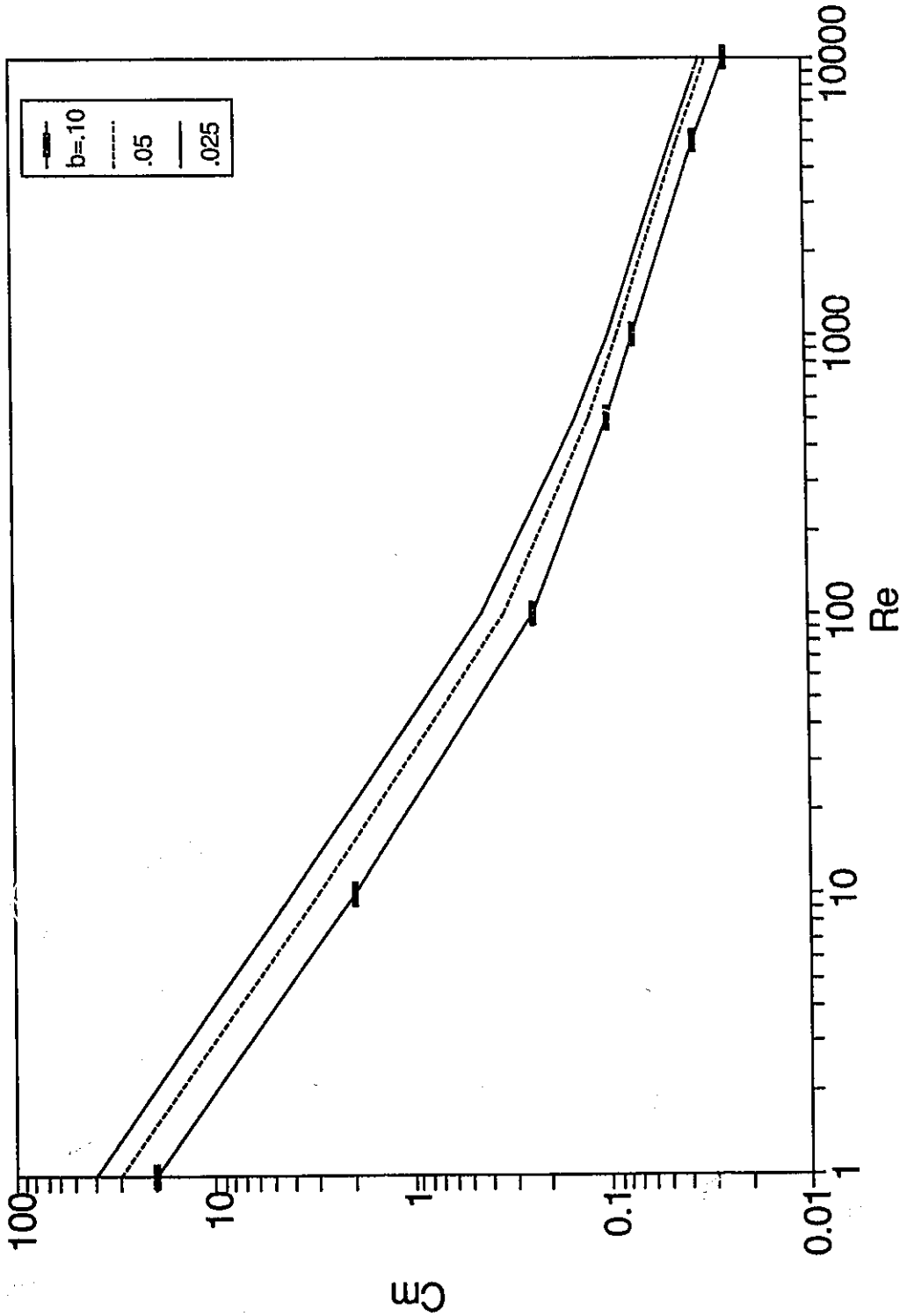

Figure 4.60 Torque Coefficient for $\delta = 0.75$

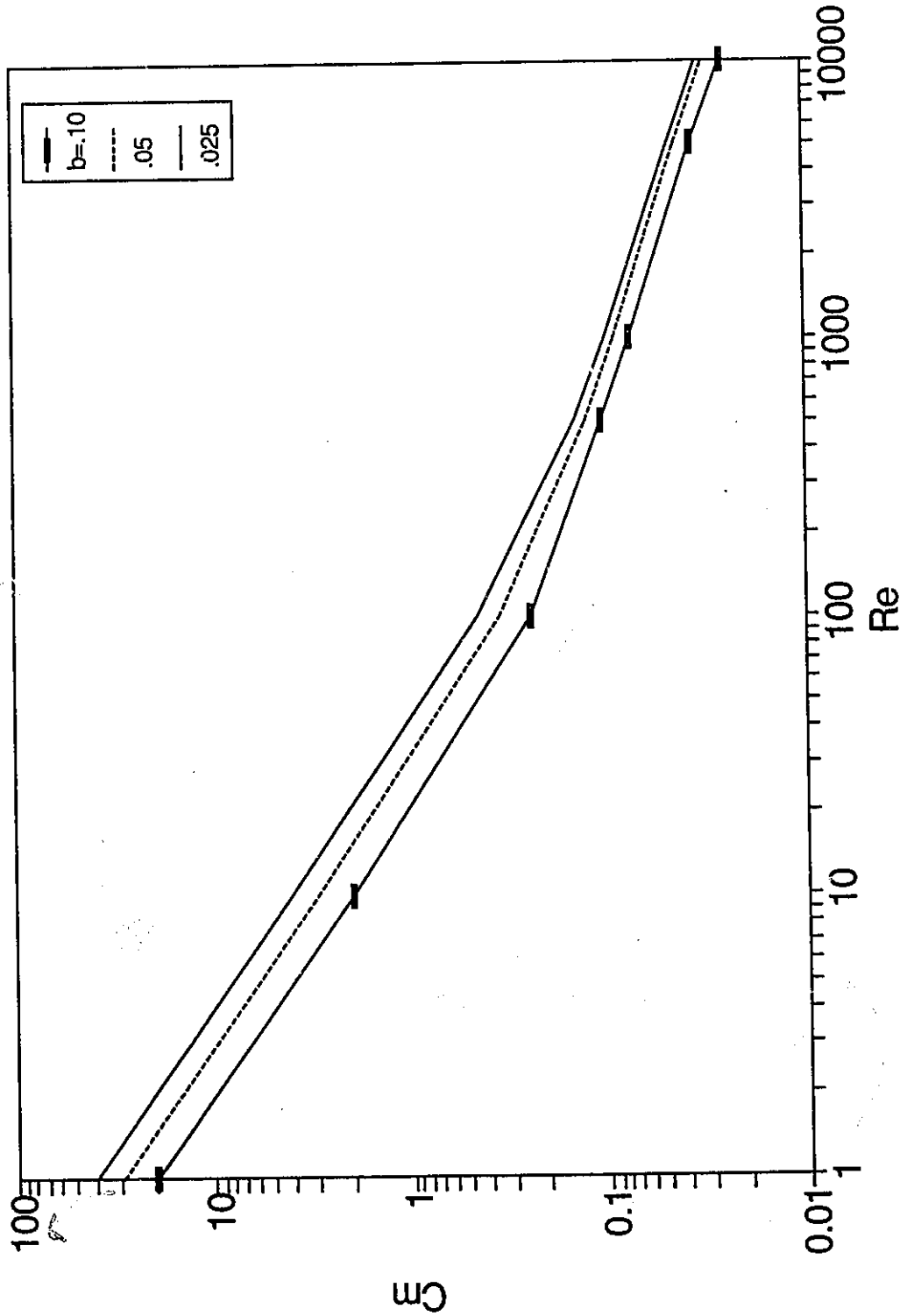

Figure 4.61 Torque Coefficient for $\delta = 1.00$

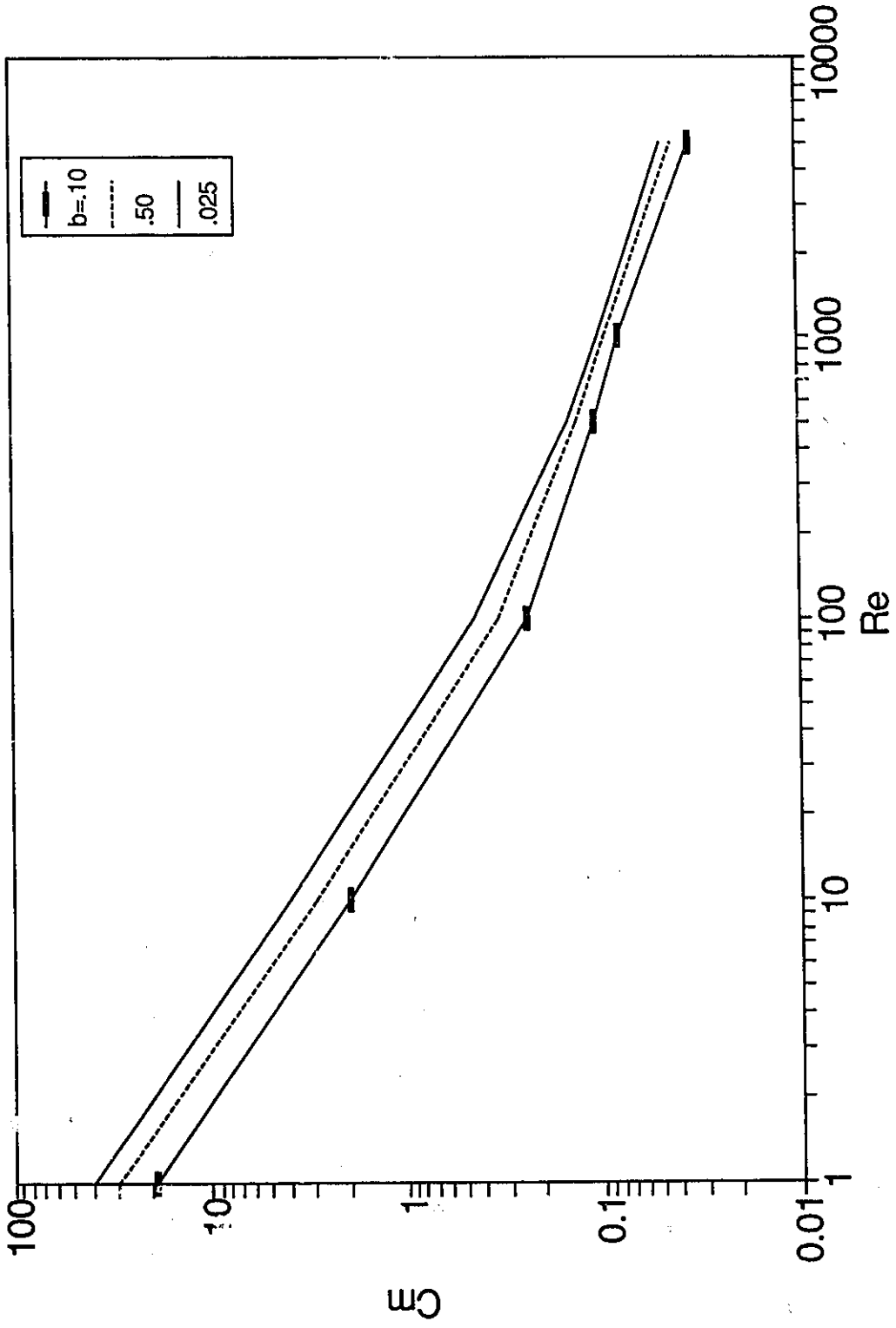

Figure 4.62 Torque Coefficient for $\delta = 1.50$

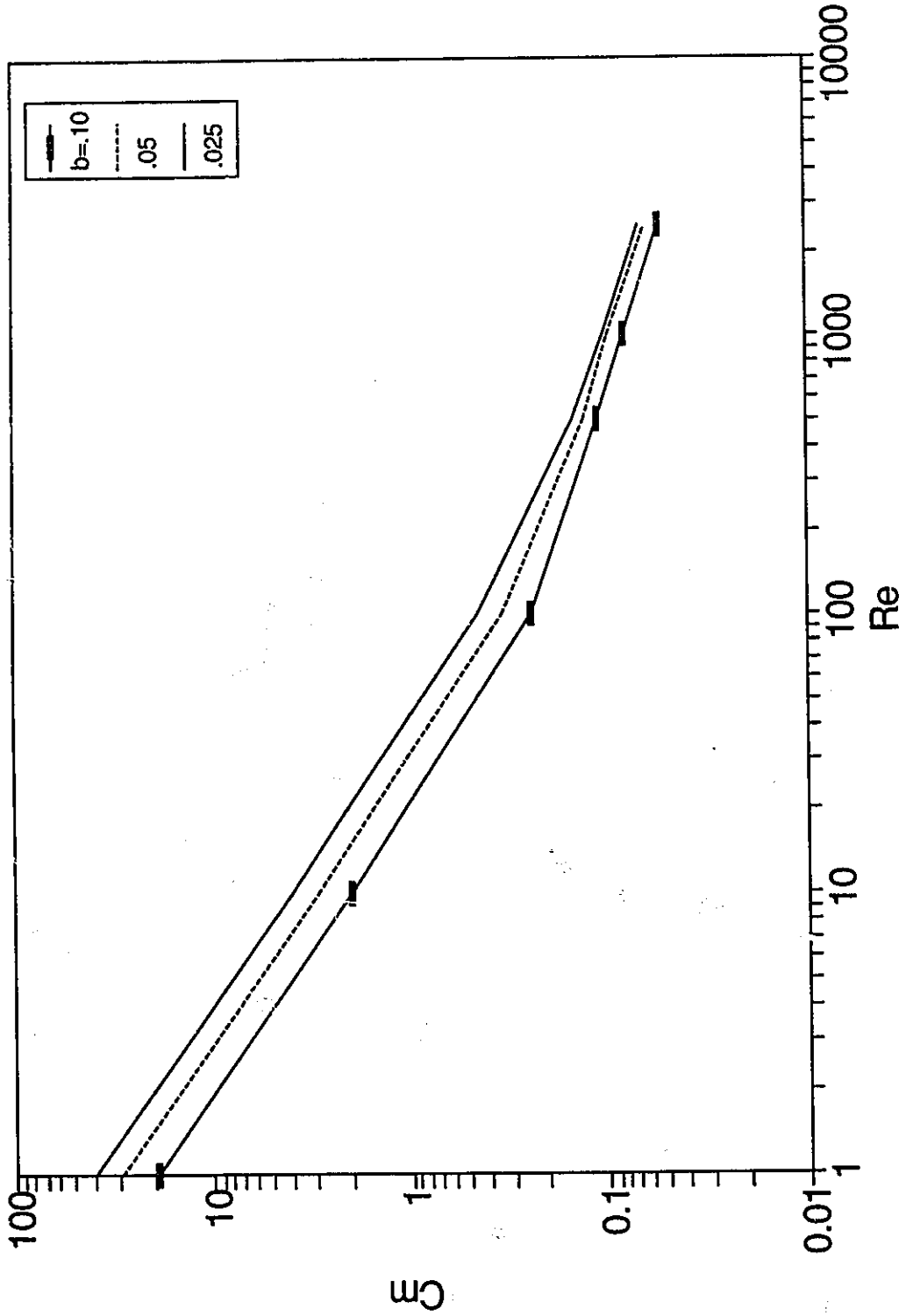

Figure 4.63 Torque Coefficient for $\delta = 2.00$

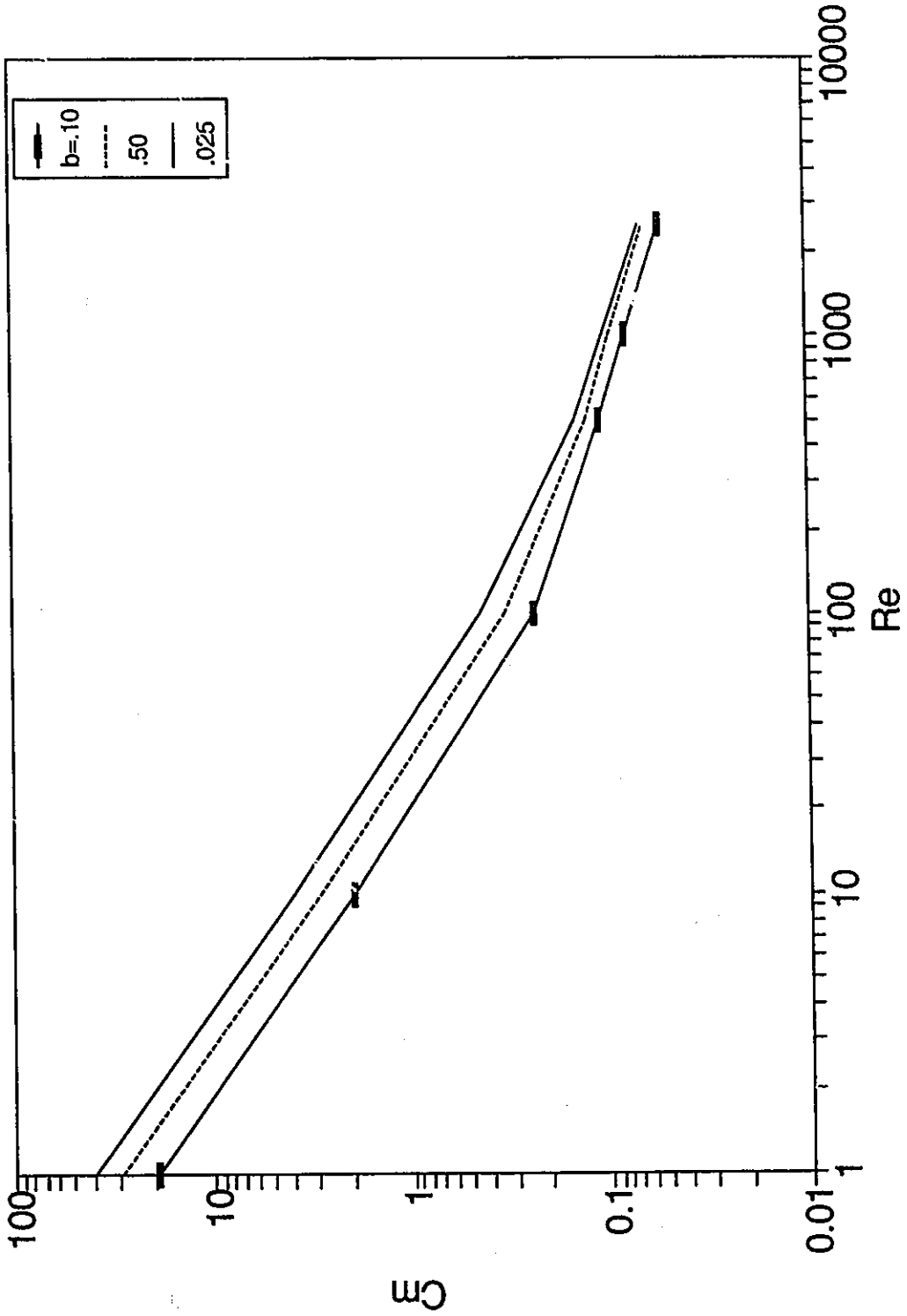

Figure 4.64 Torque Coefficient for $\delta = 2.50$

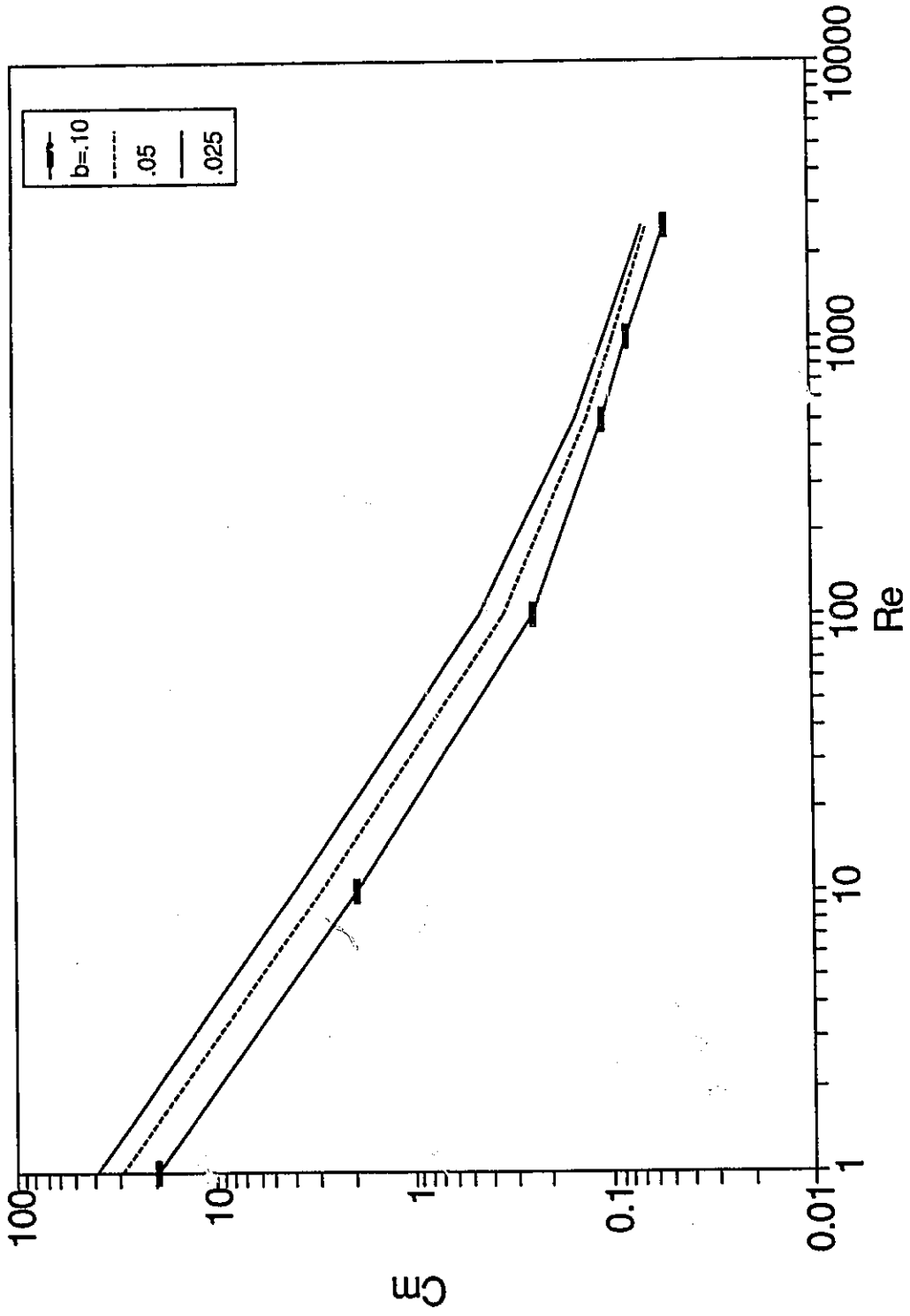

Figure 4.65 Torque Coefficient for $\delta = 3.00$

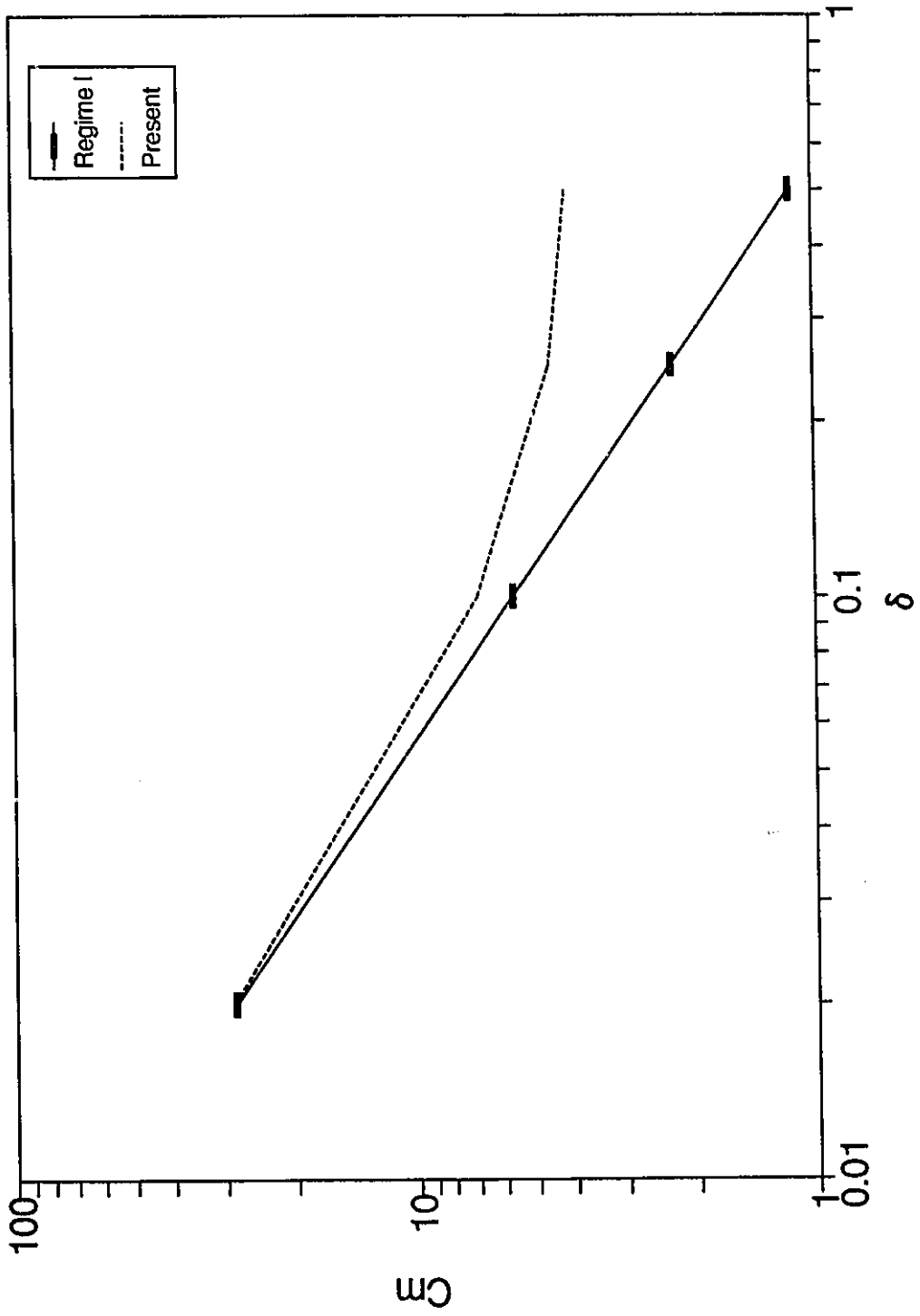

Figure 4.66 Comparison of Torque Coefficient by Equation (4.18) with Computational Method for Re = 10 and b = 0.025

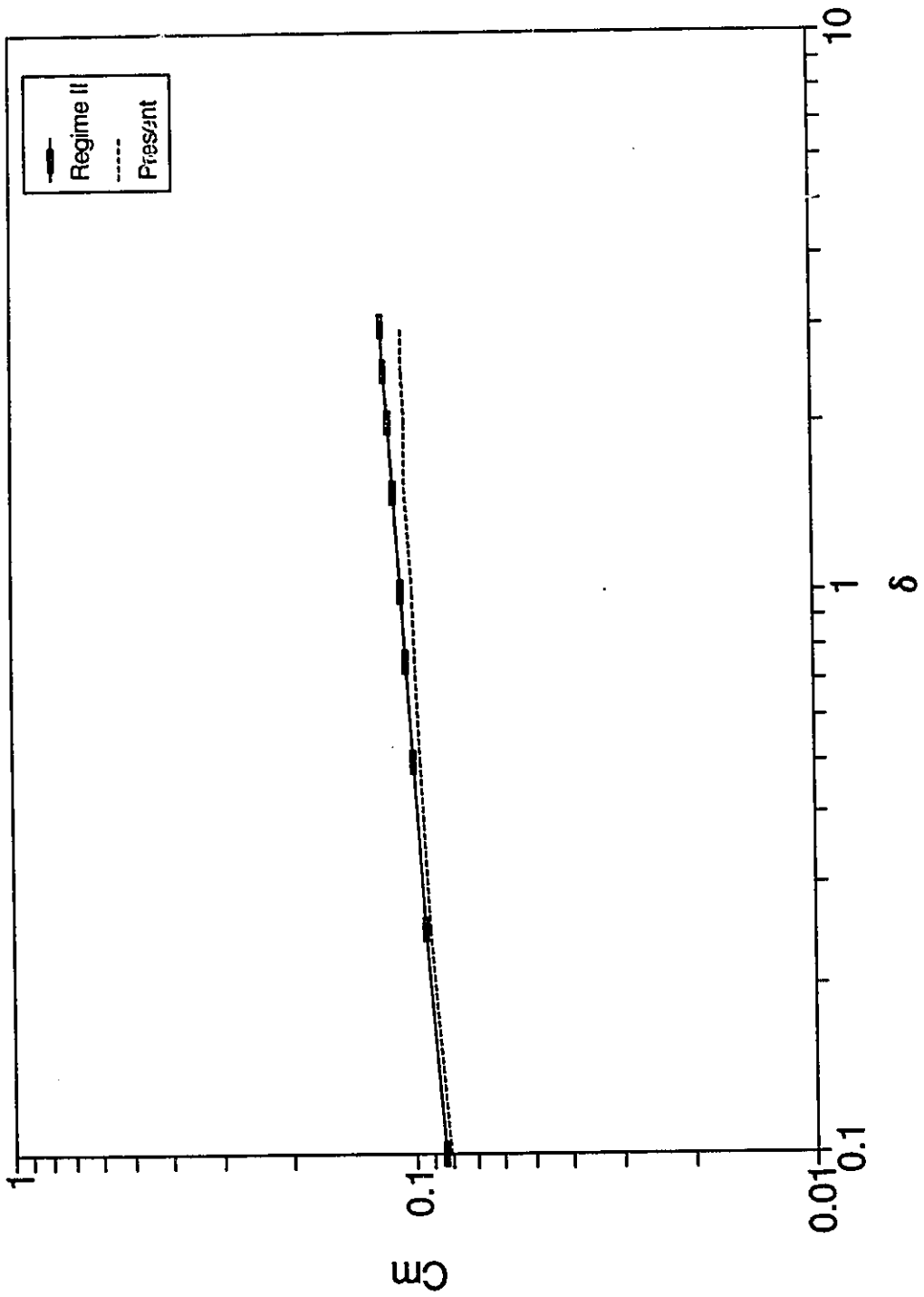

Figure 4.67 Comparison of Torque Coefficient by Equation (4.19) with Computational Method for Re = 10 000 and b = 0.025

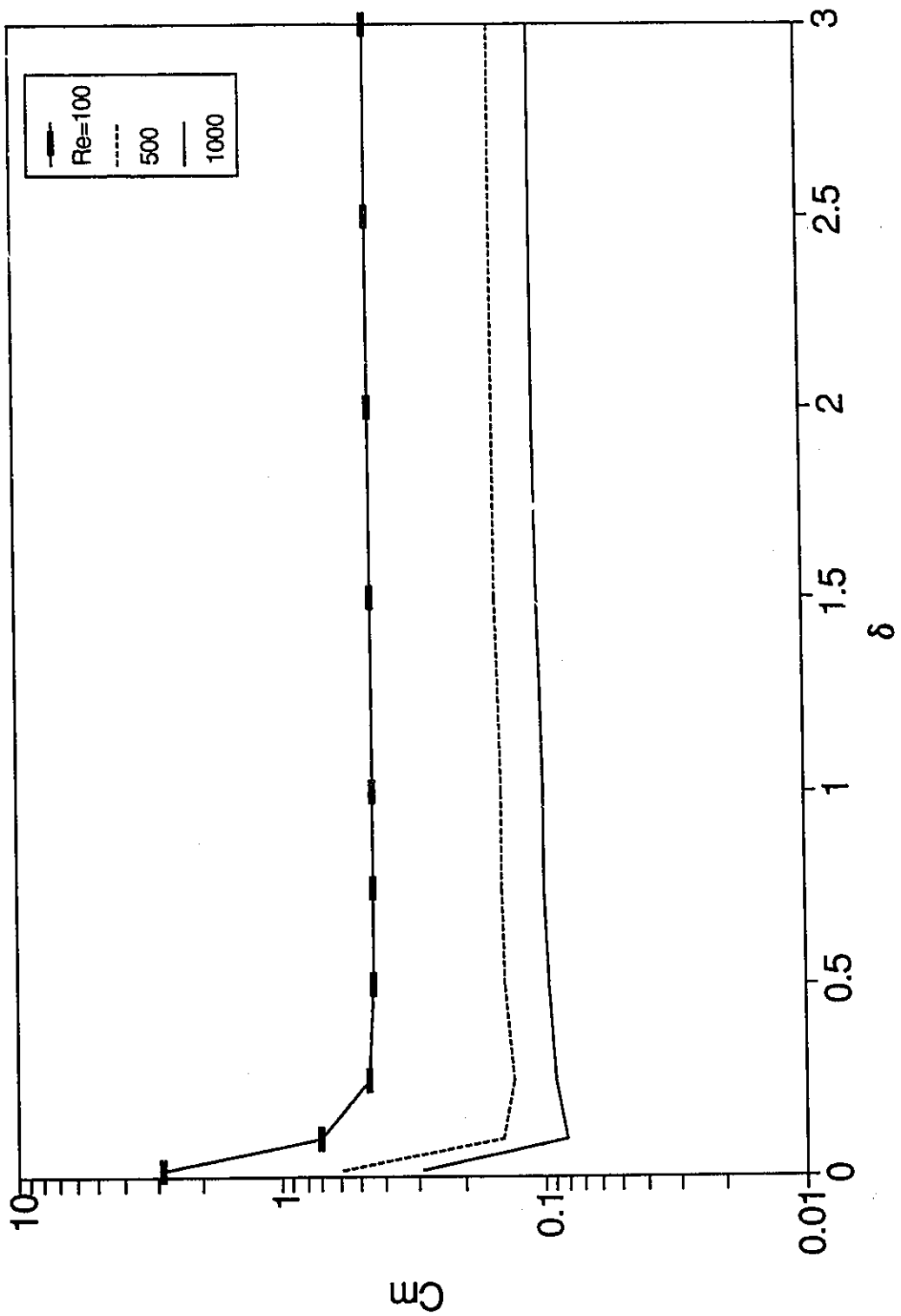

Figure 4.68 Torque Coefficient as a Function of δ for b = 0.025

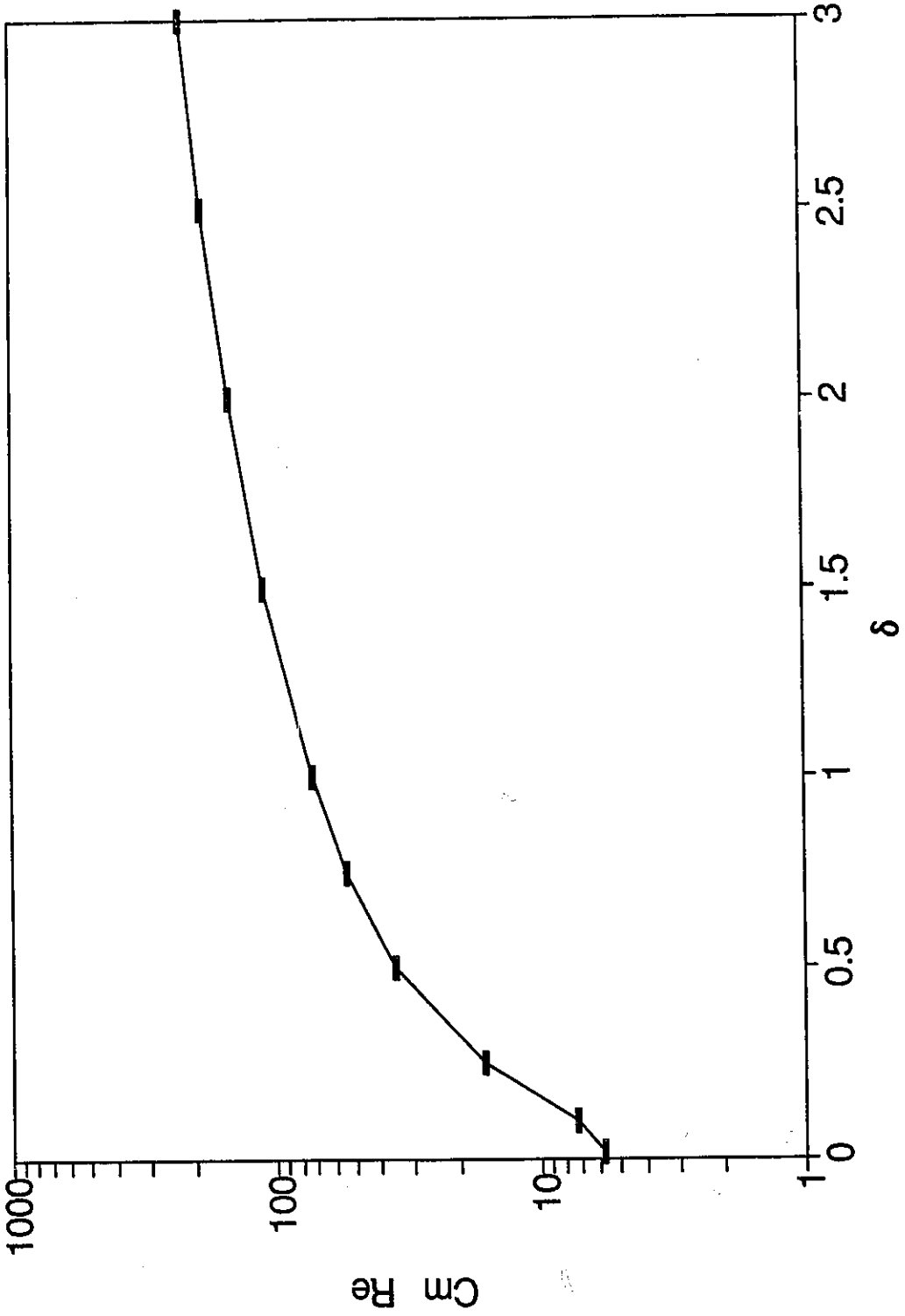

Figure 4.69 Torque Integral as a Function of δ for Re = 500 and b = 0.025

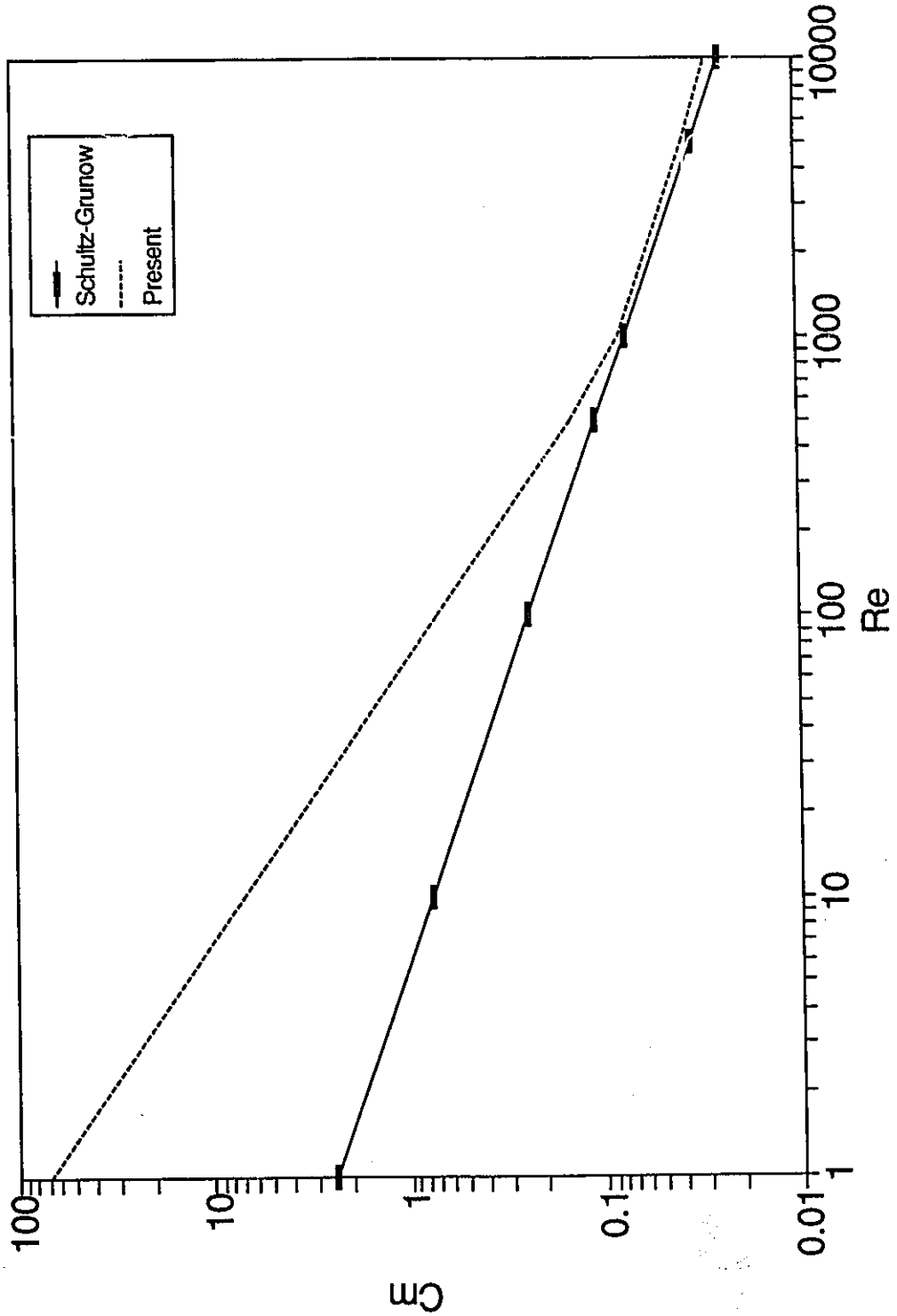

Figure 4.70 Comparison of the Schultz-Grunow Correlation, Equation (4.21), with the Computational Method for $\delta = 0.10$ and $b = 0.025$

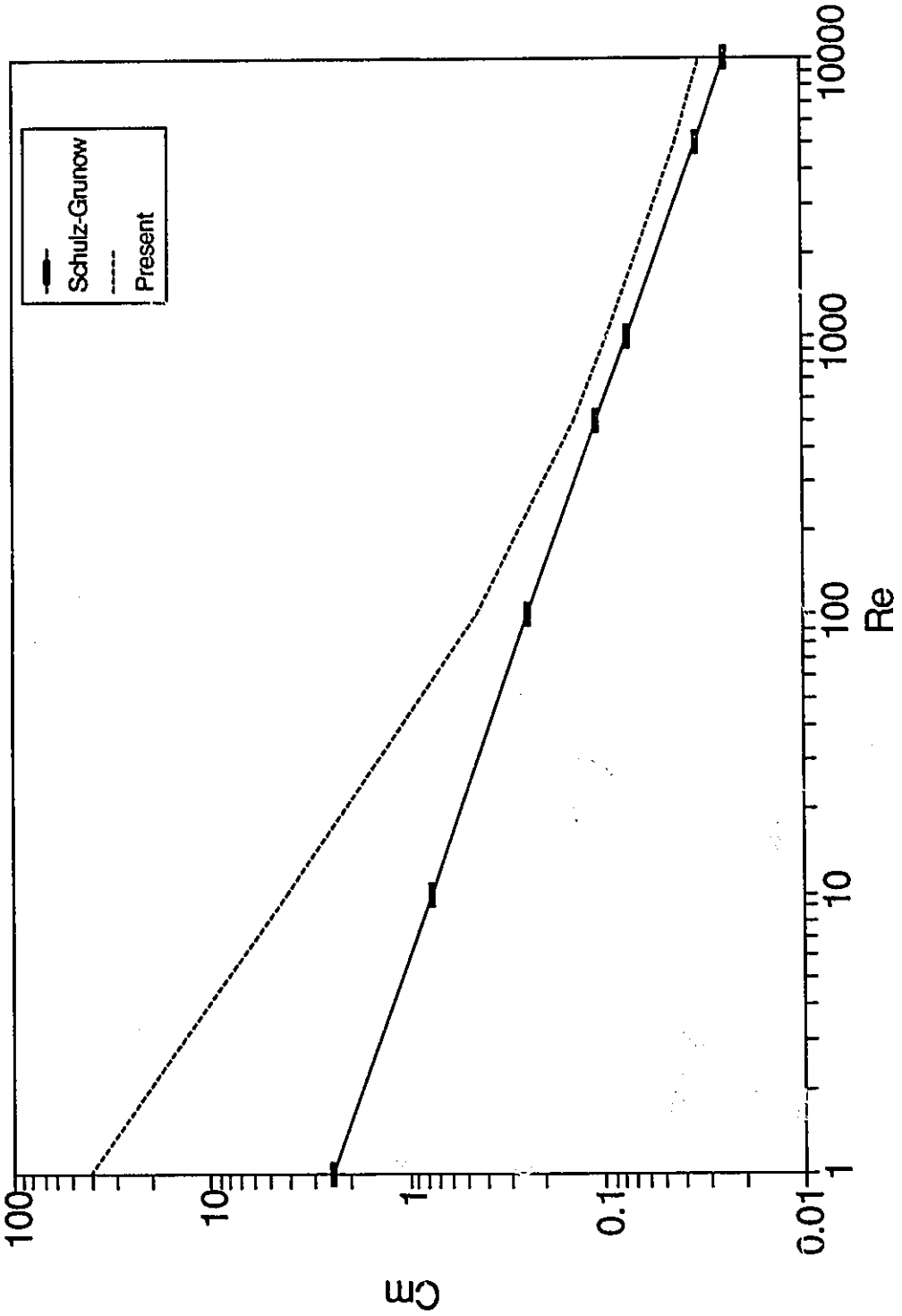

Figure 4.71 Comparison of the Schultz-Grunow Correlation, Equation (4.21), with the Computational Method for $\delta = 0.50$ and $b = 0.025$

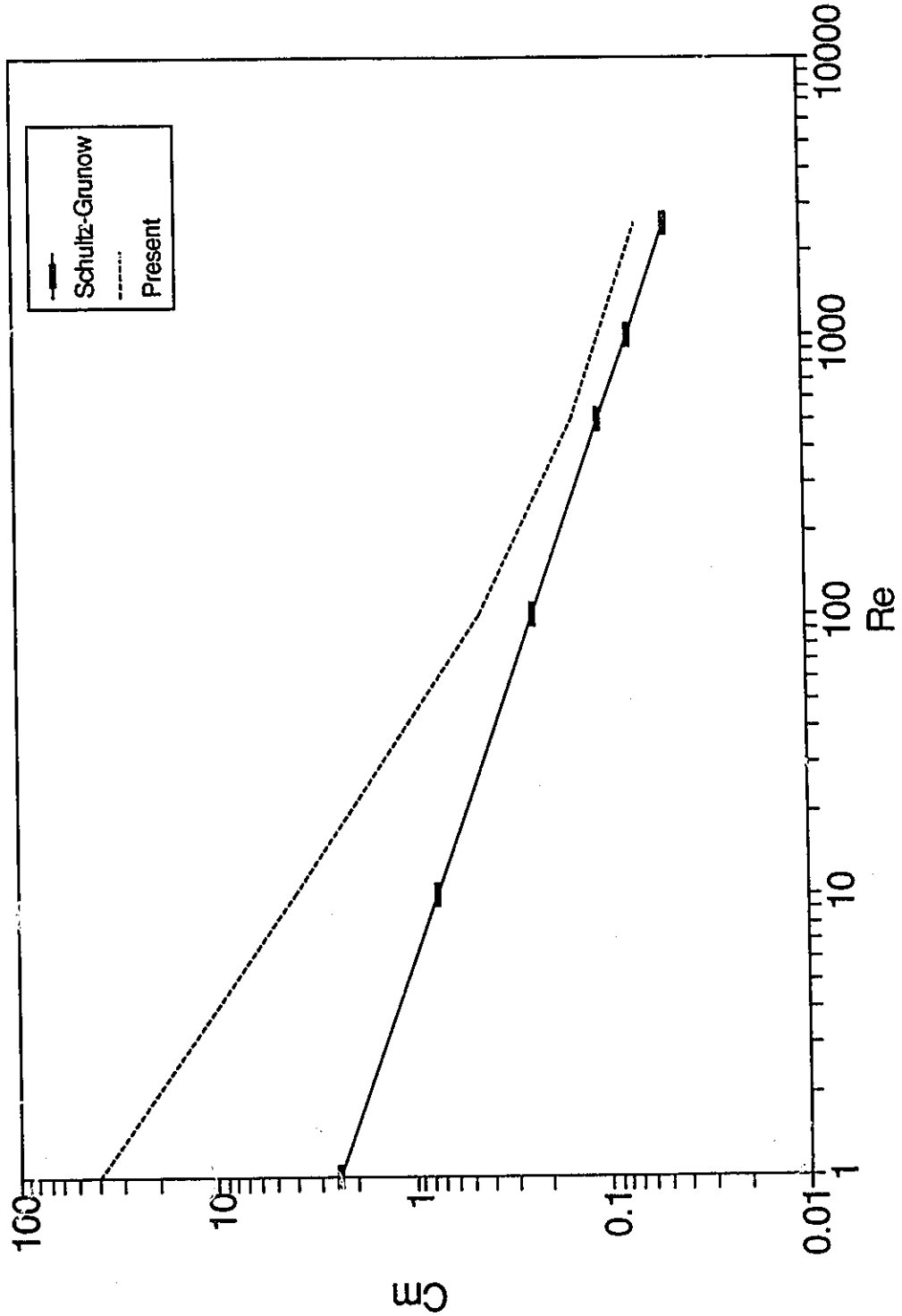

Figure 4.72 Comparison of the Schultz-Grunow Correlation, Equation (4.21), with the Computational Method for $\delta = 3.00$ and $b = 0.025$

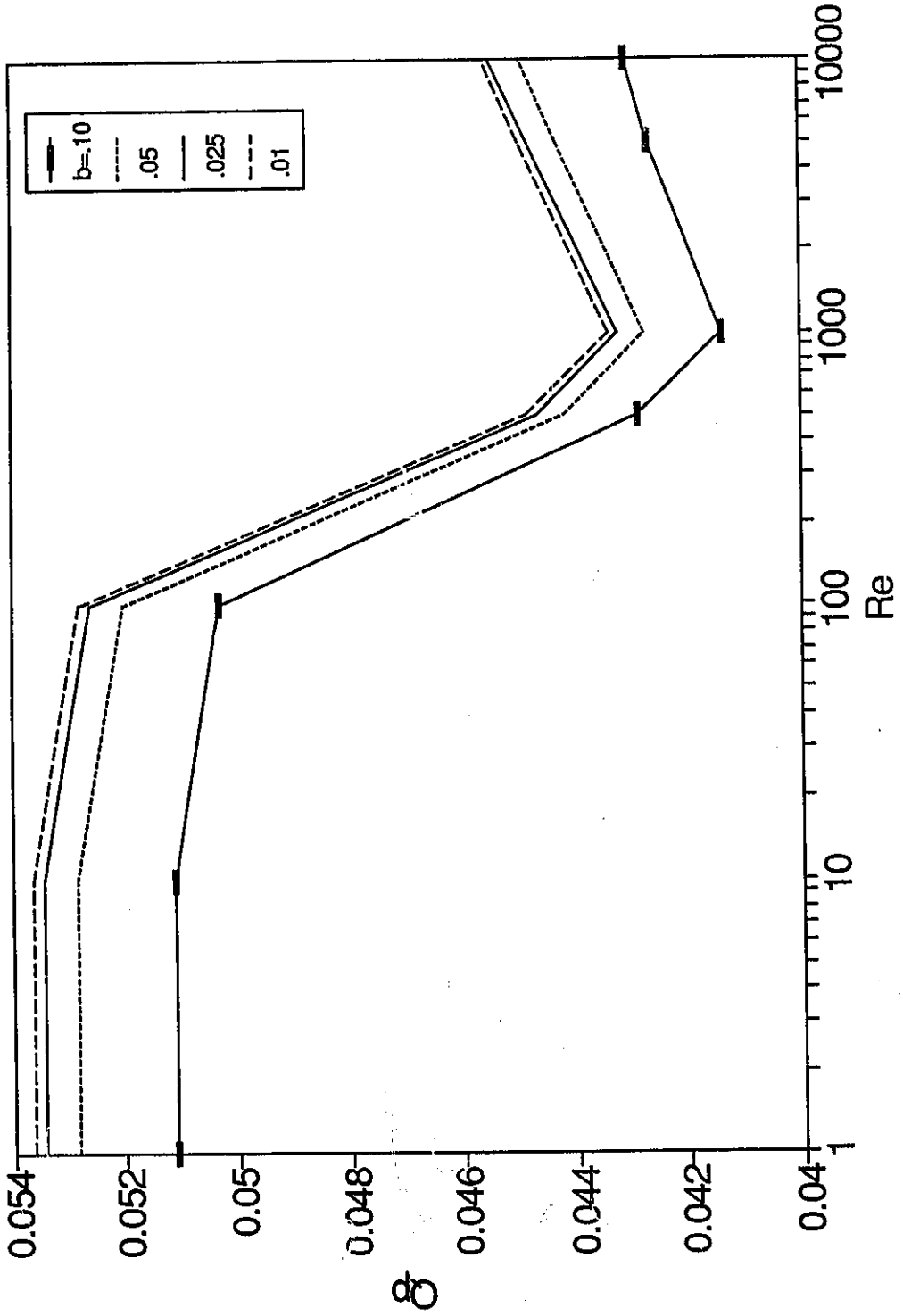

Figure 4.73 Variation of the Primary Volumetric Flow Rate with the Gap for $\delta = 0.25$

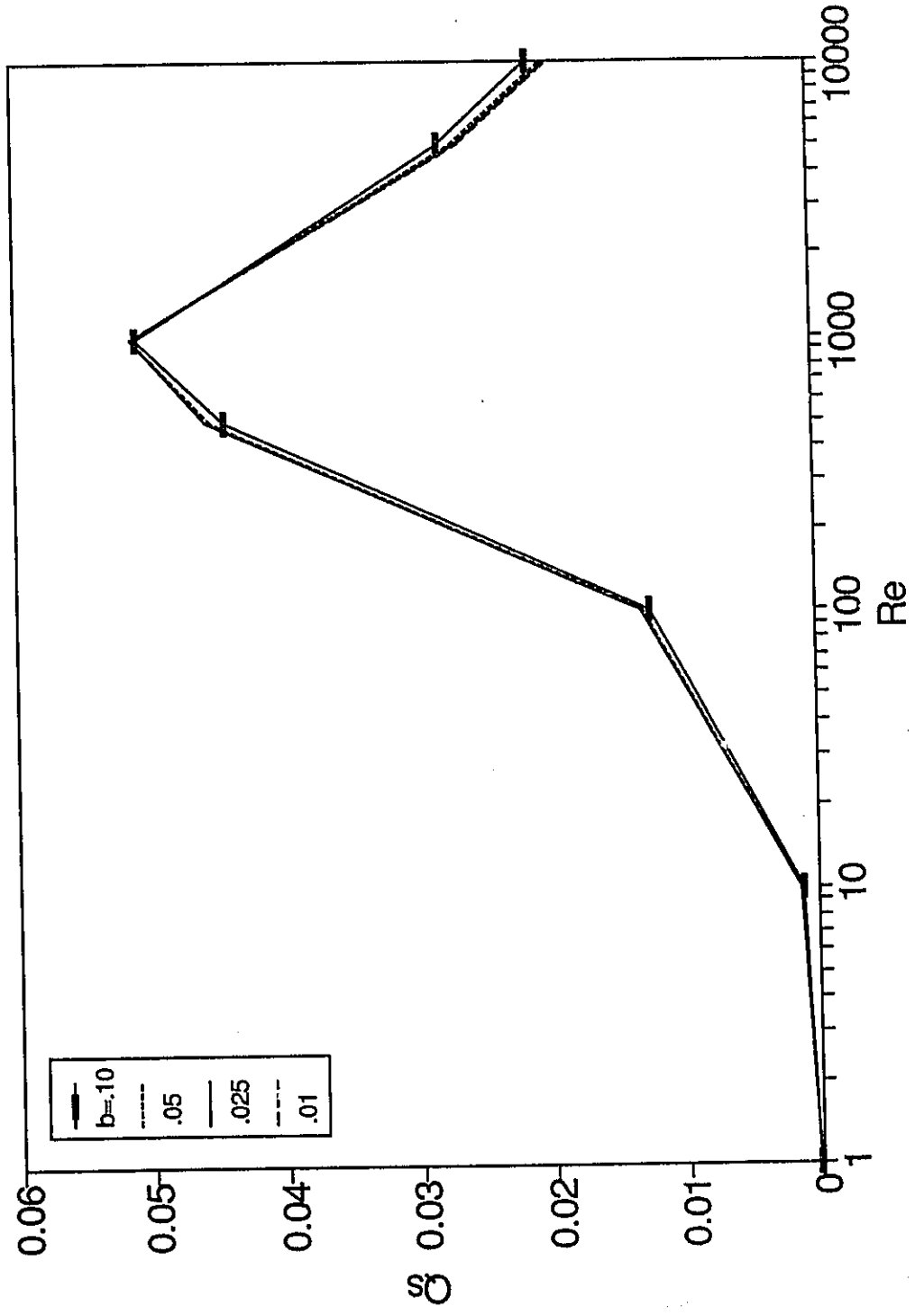

Figure 4.74 Variation of the Secondary Volumetric Flow Rate with the Gap for $\delta = 0.25$

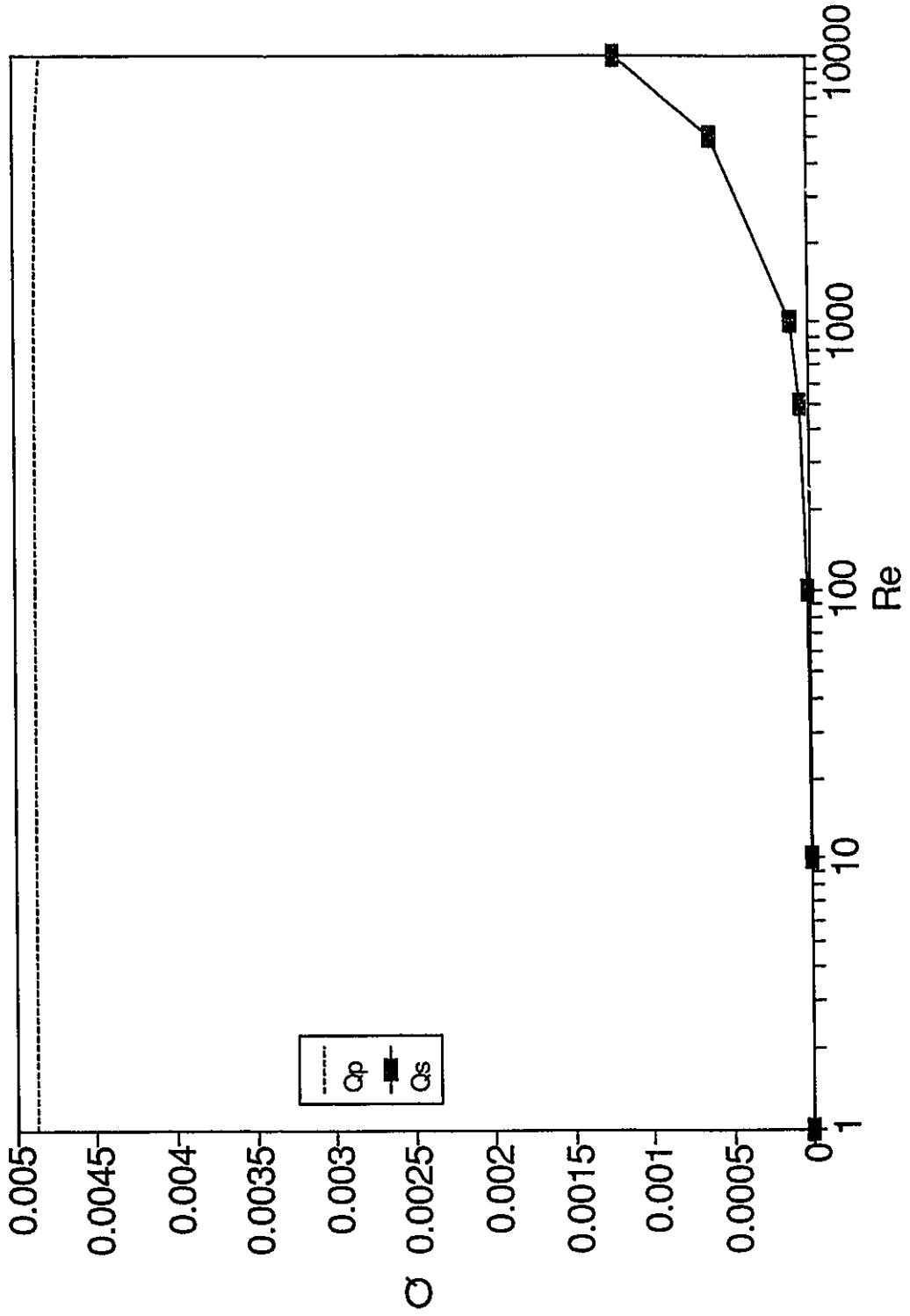

Figure 4.75 Volumetric Flow Rates for $\delta = 0.02$ and $b = 0.025$

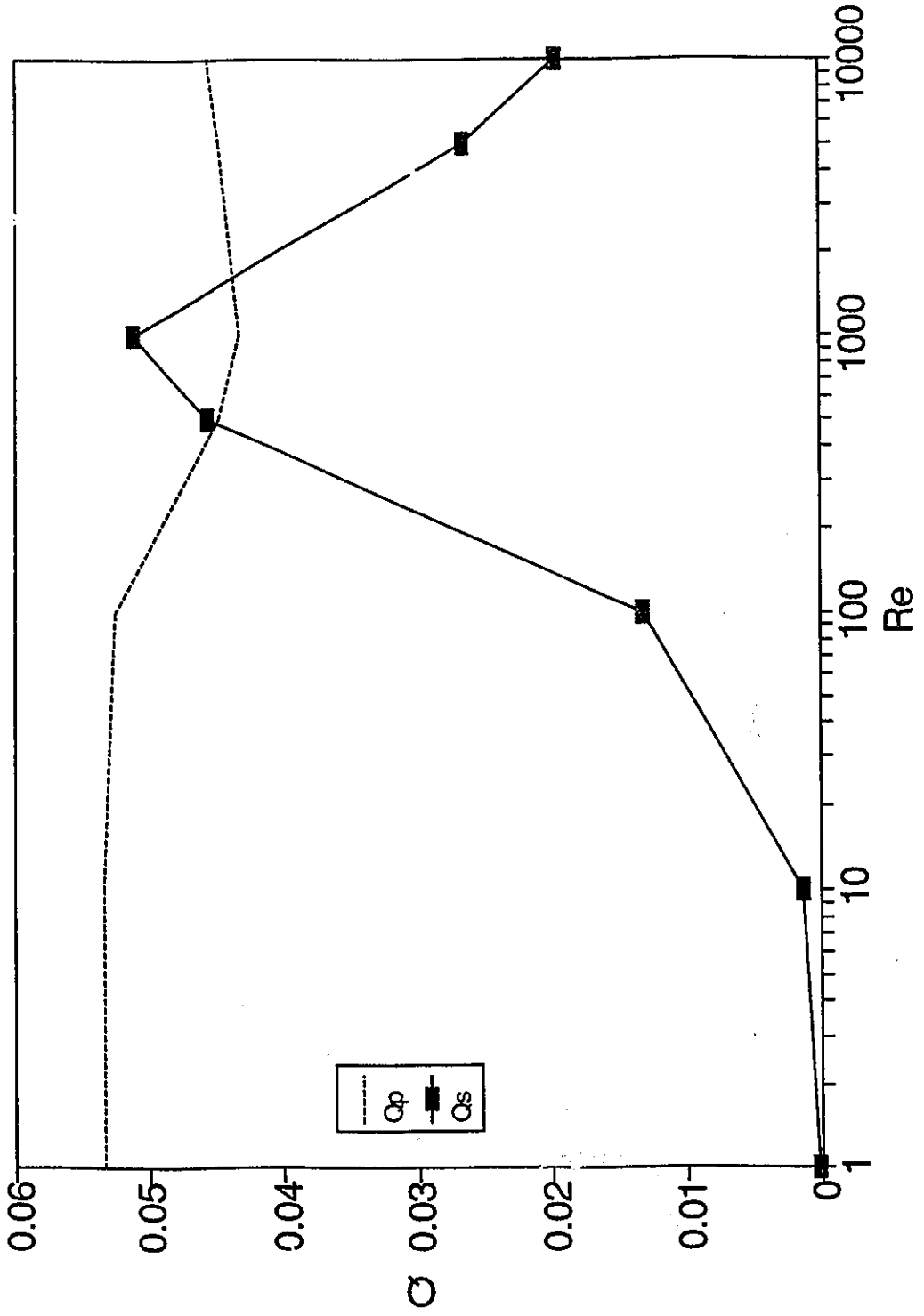

Figure 4.76 Volumetric Flow Rates for $\delta = 0.10$ and $b = 0.025$

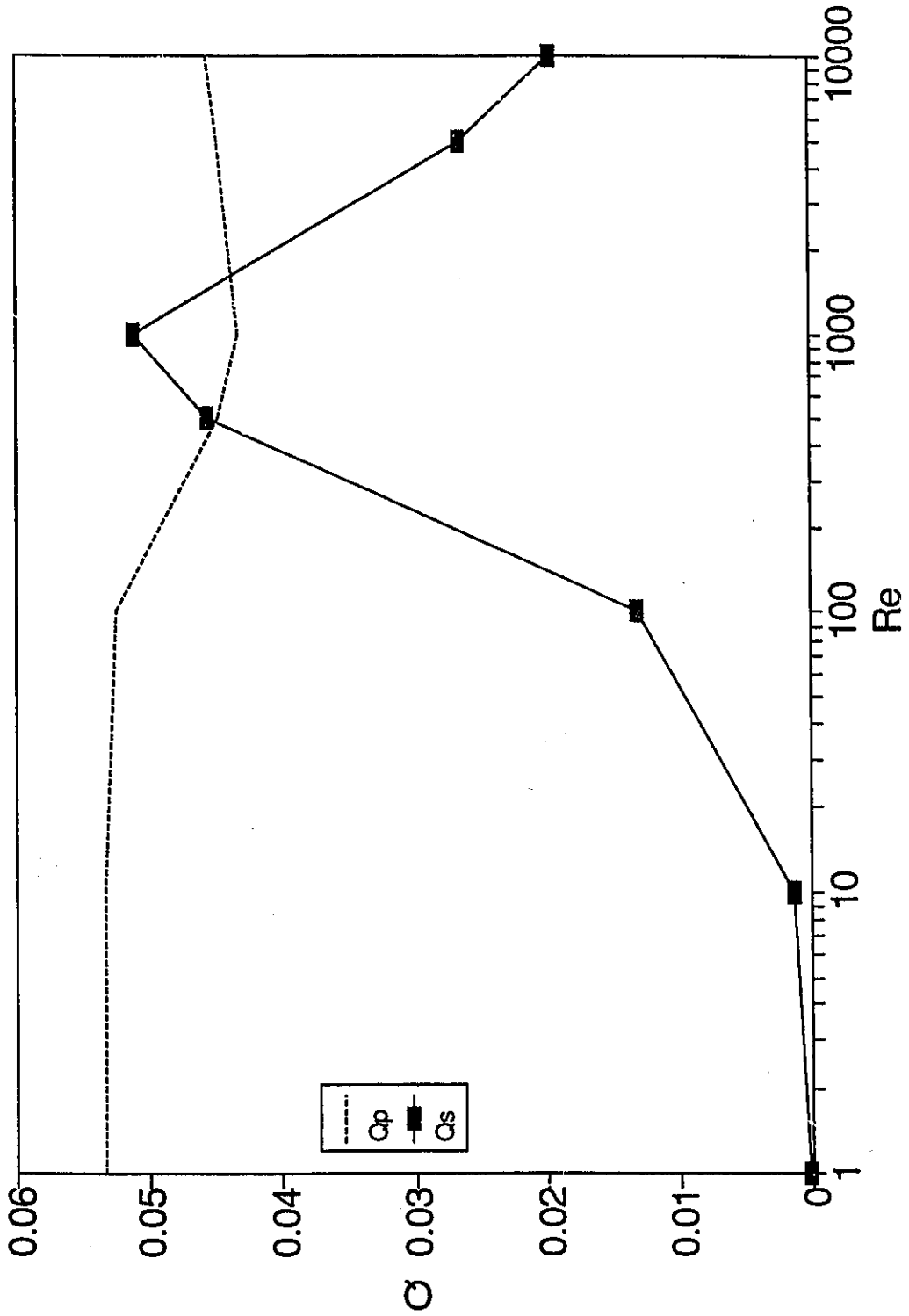

Figure 4.77 Volumetric Flow Rates for $\delta = 0.25$ and $b = 0.025$

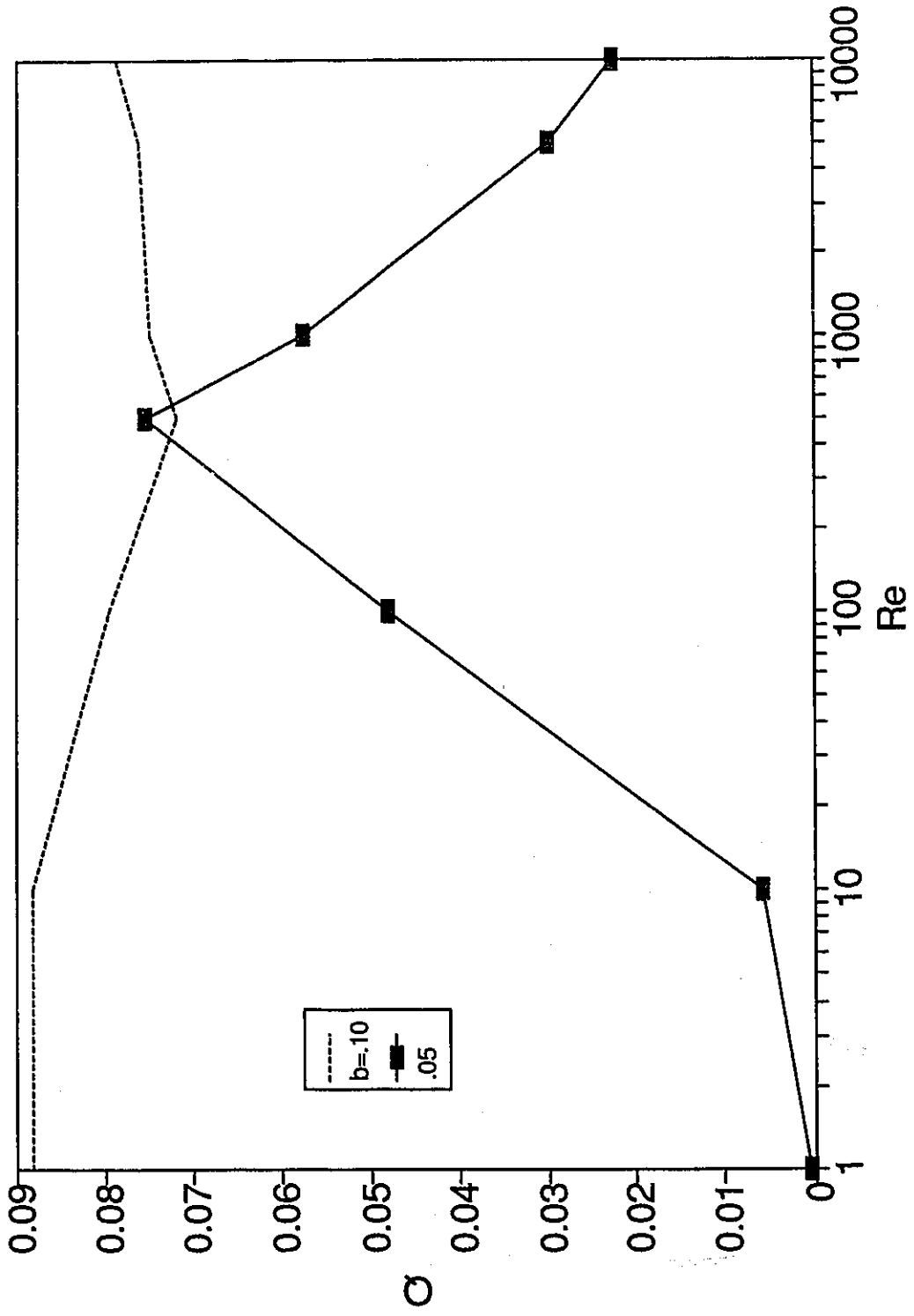

Figure 4.78 Volumetric Flow Rates for $\delta = 0.50$ and $b = 0.025$

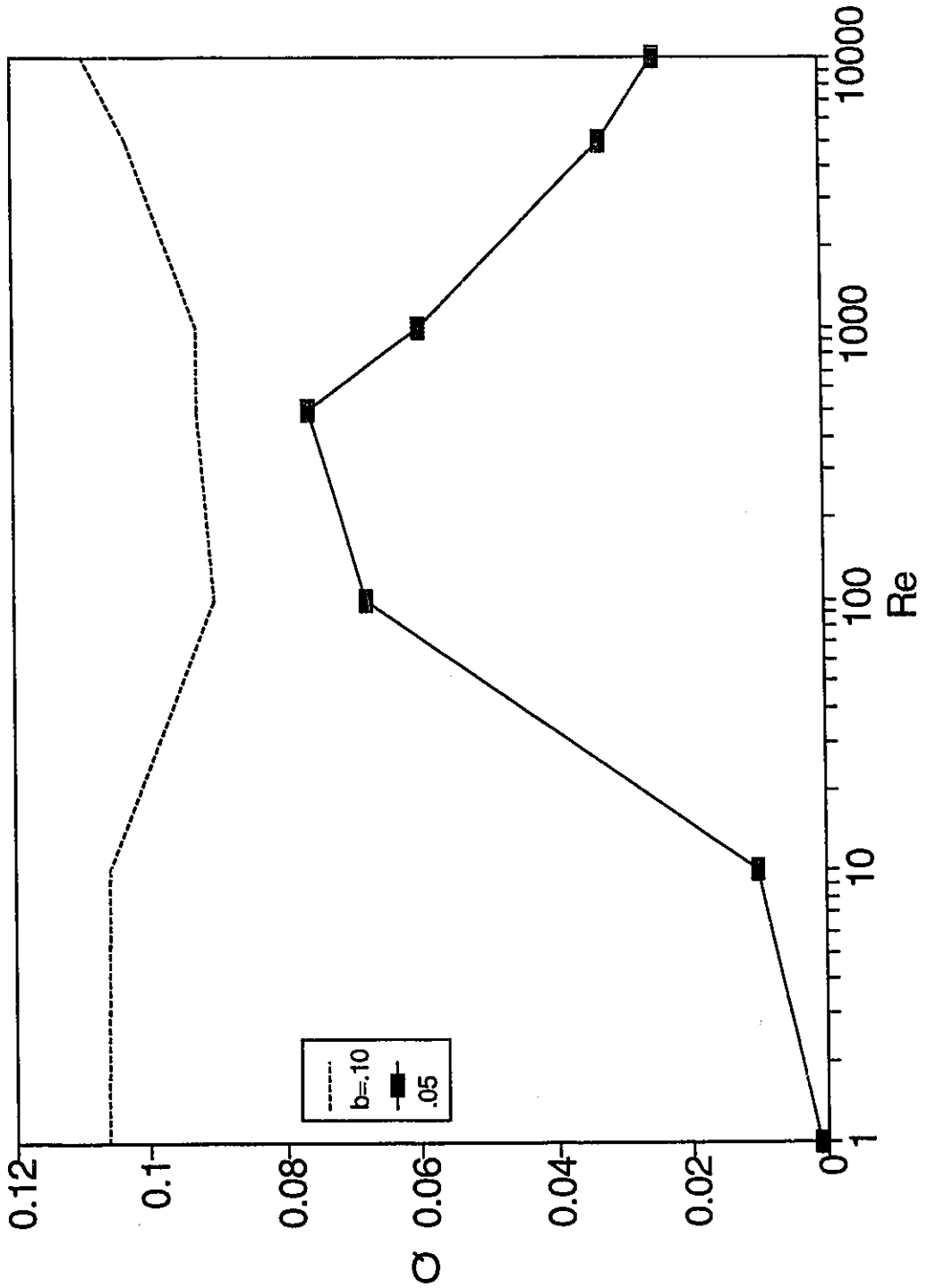

Figure 4.79 Volumetric Flow Rates for $\delta = 0.75$ and $b = 0.025$

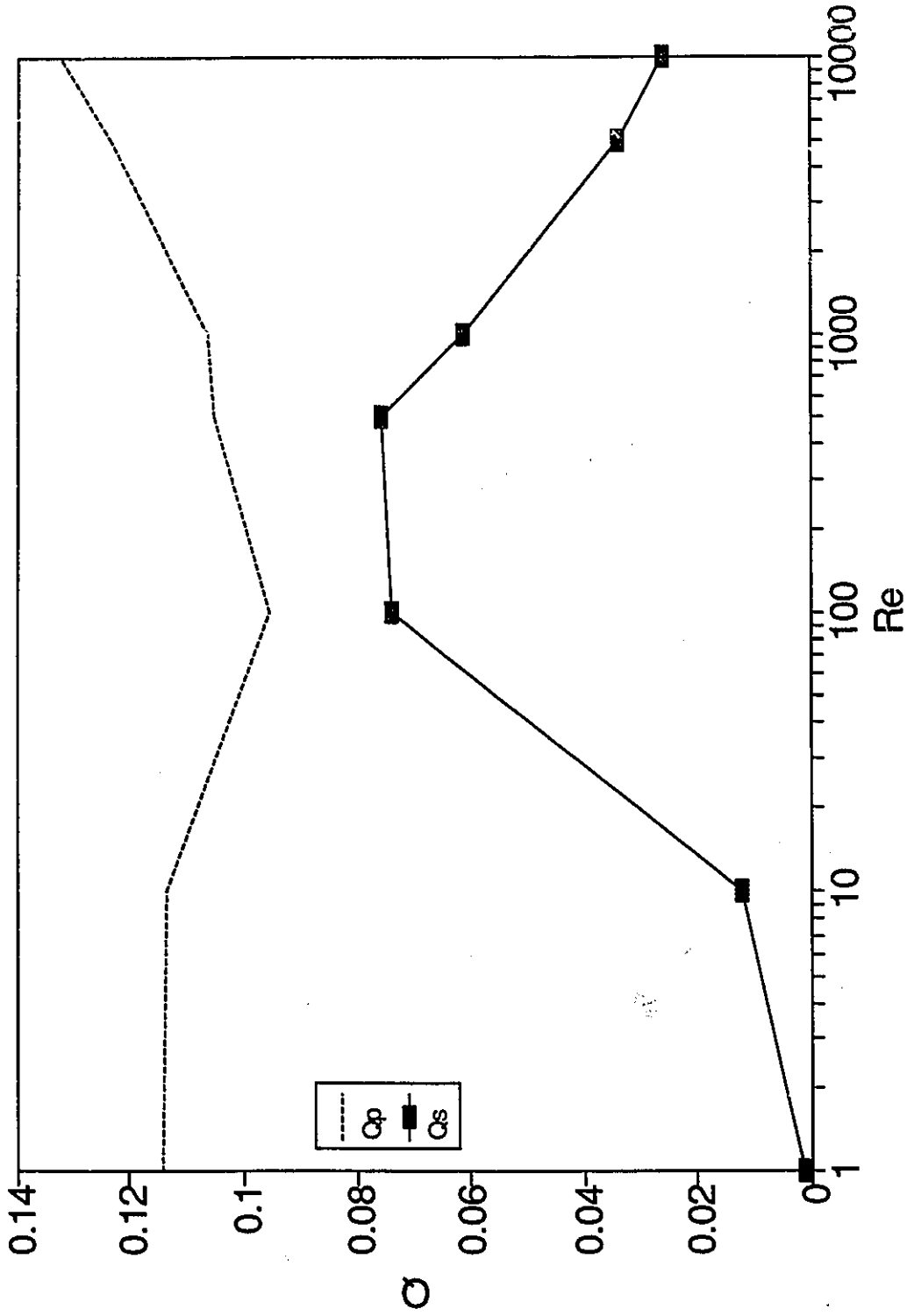

Figure 4.80 Volumetric Flow Rates for $\delta = 1.00$ and $b = 0.025$

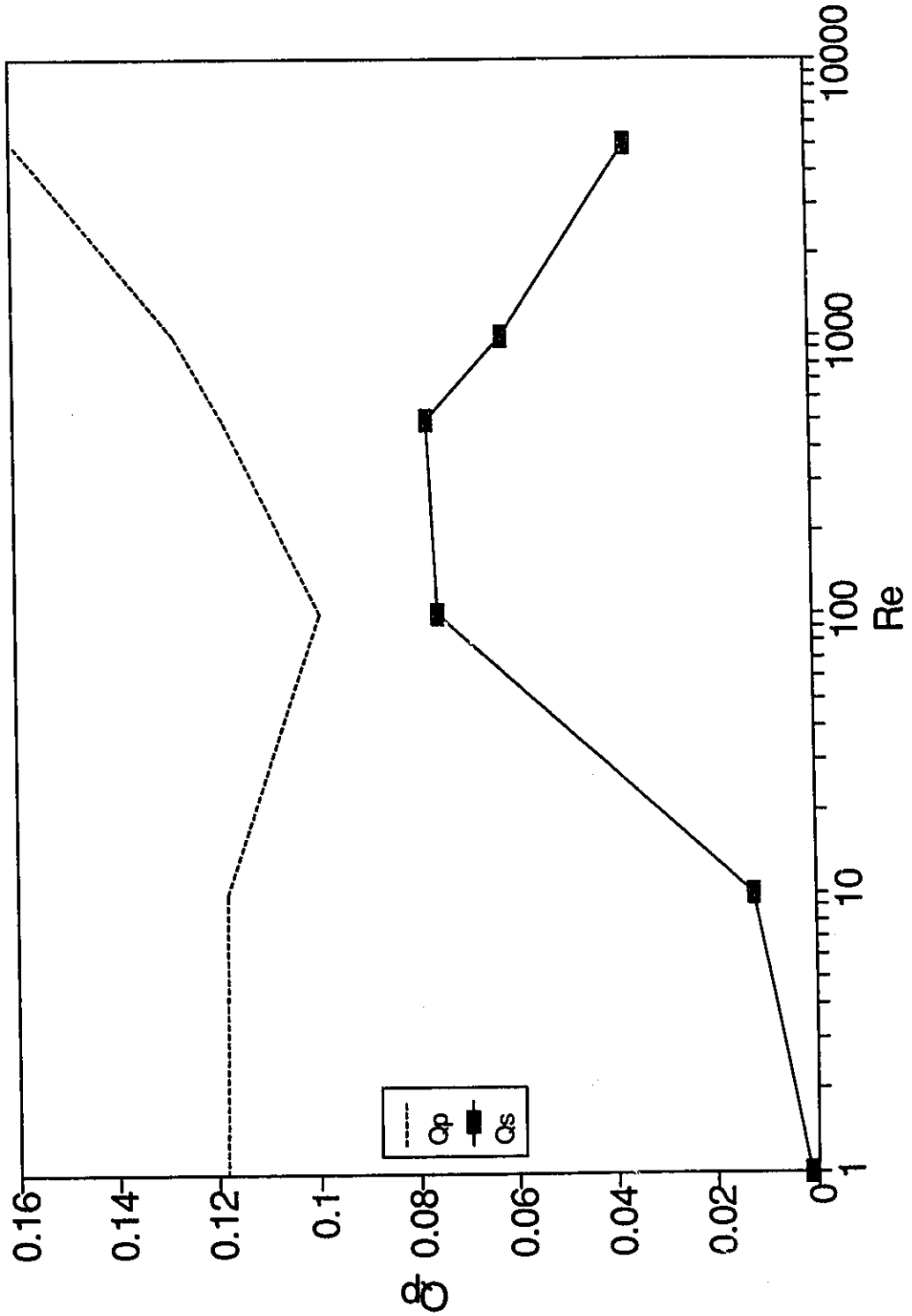

Figure 4.81 Volumetric Flow Rates for $\delta = 1.50$ and $b = 0.025$

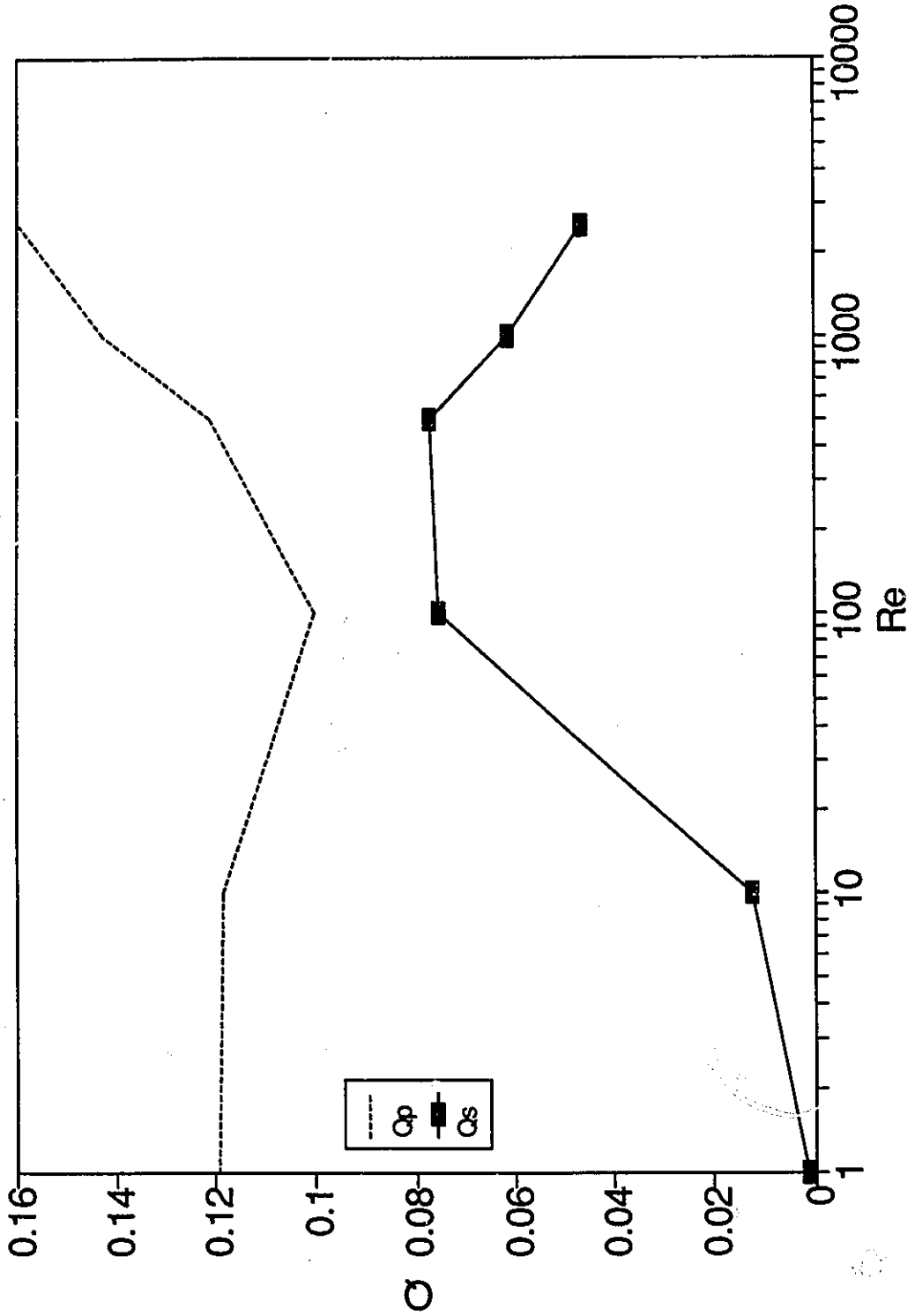

Figure 4.82 Volumetric Flow Rates for $\delta = 2.00$ and $b = 0.025$

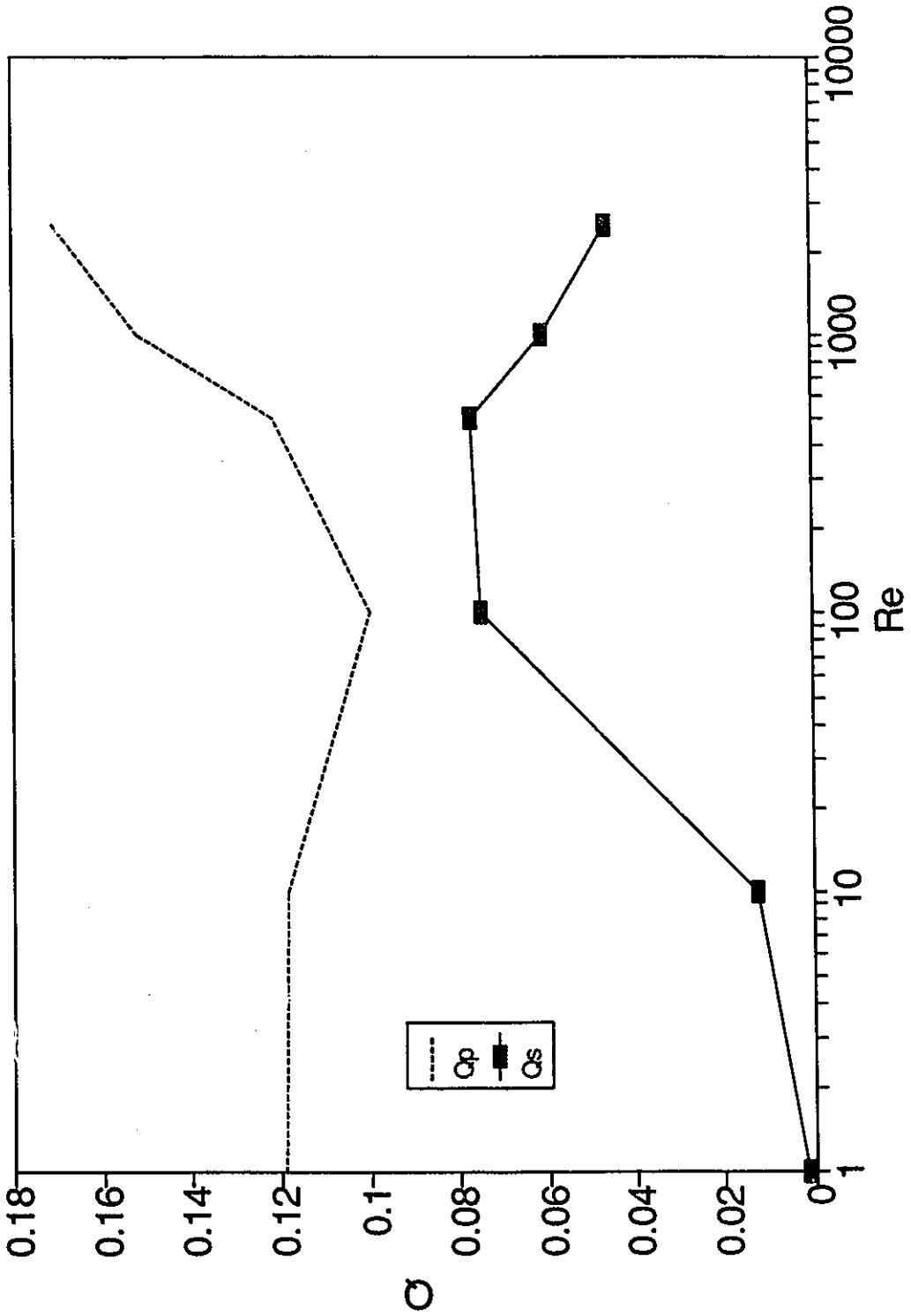

Figure 4.83 Volumetric Flow Rates for δ = 2.50 and b = 0.025

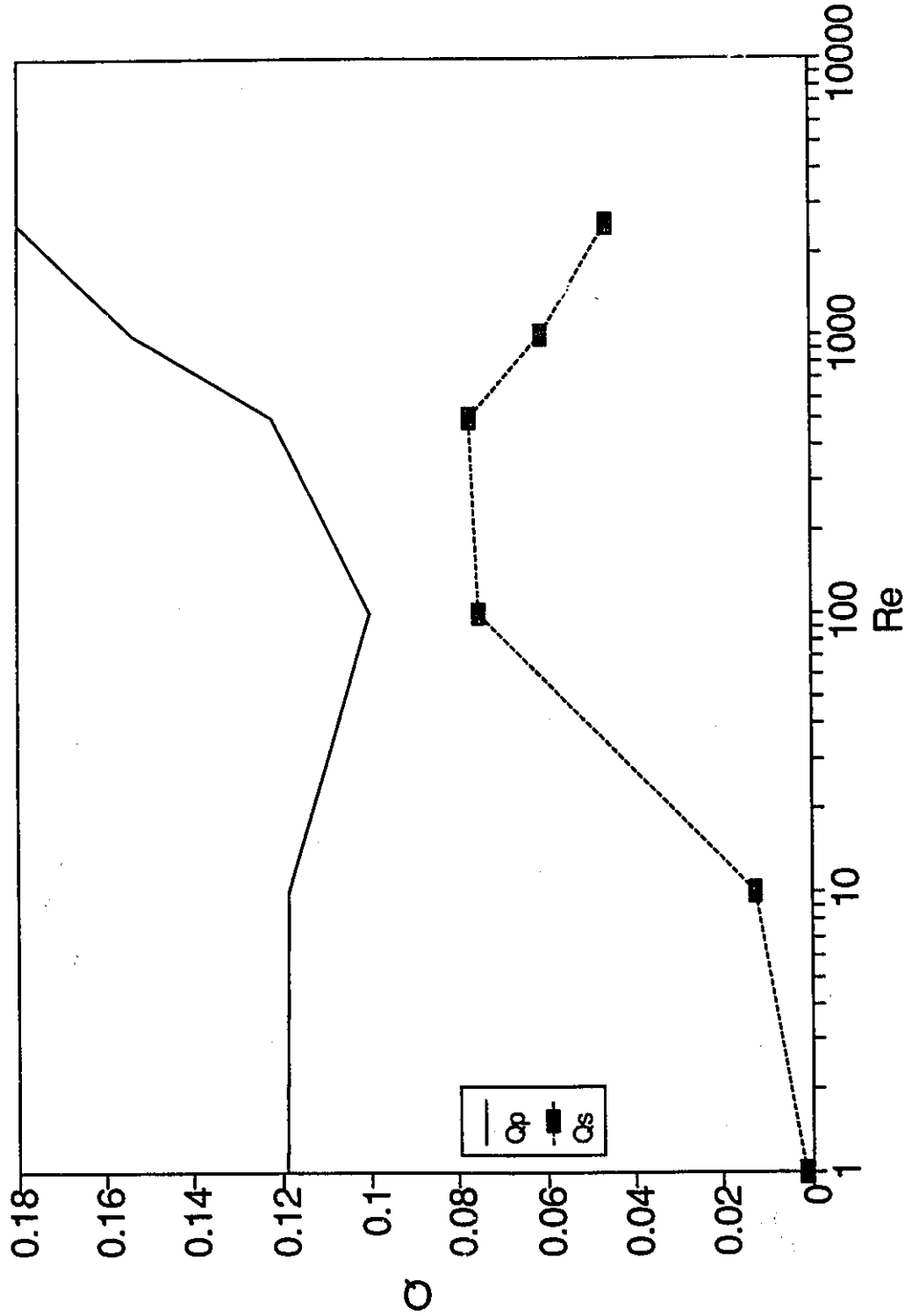

Figure 4.84 Volumetric Flow Rates for $\delta = 3.00$ and $b = 0.025$

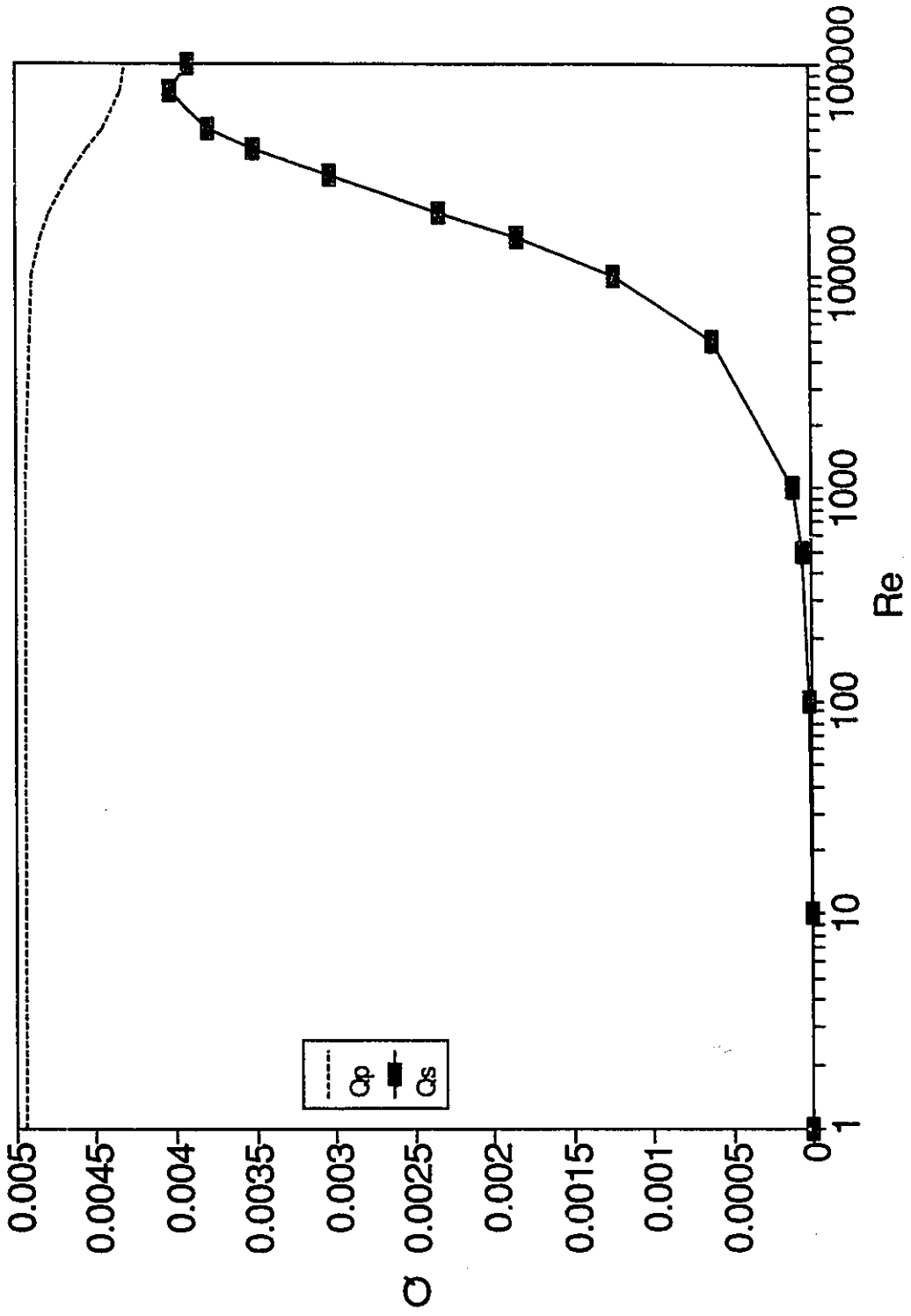

Figure 4.85 Volumetric Flow Rates for $\delta = 0.02$ and $b = 0.0063$

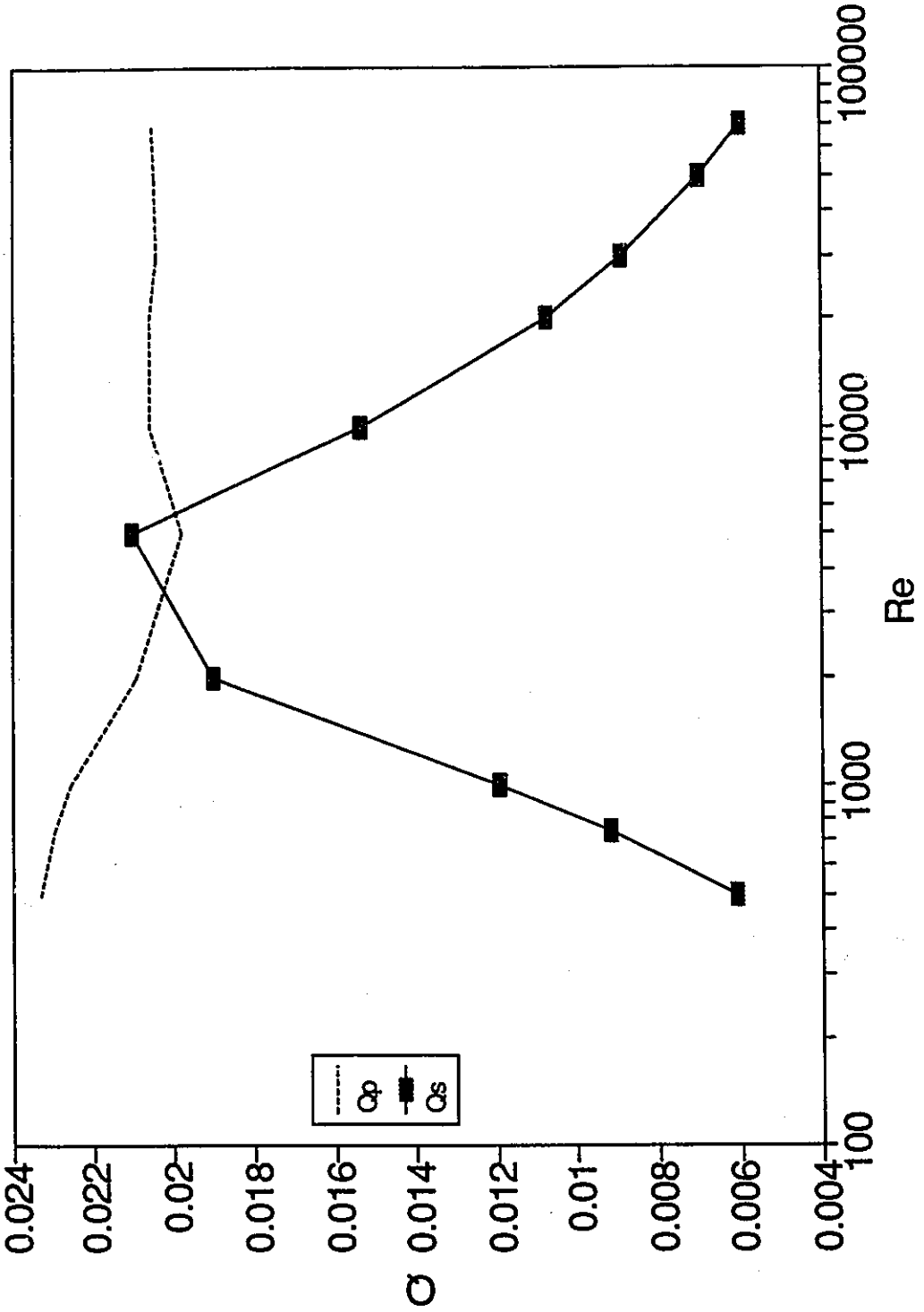

Figure 4.86 Volumetric Flow Rates for $\delta = 0.10$ and $b = 0.0063$

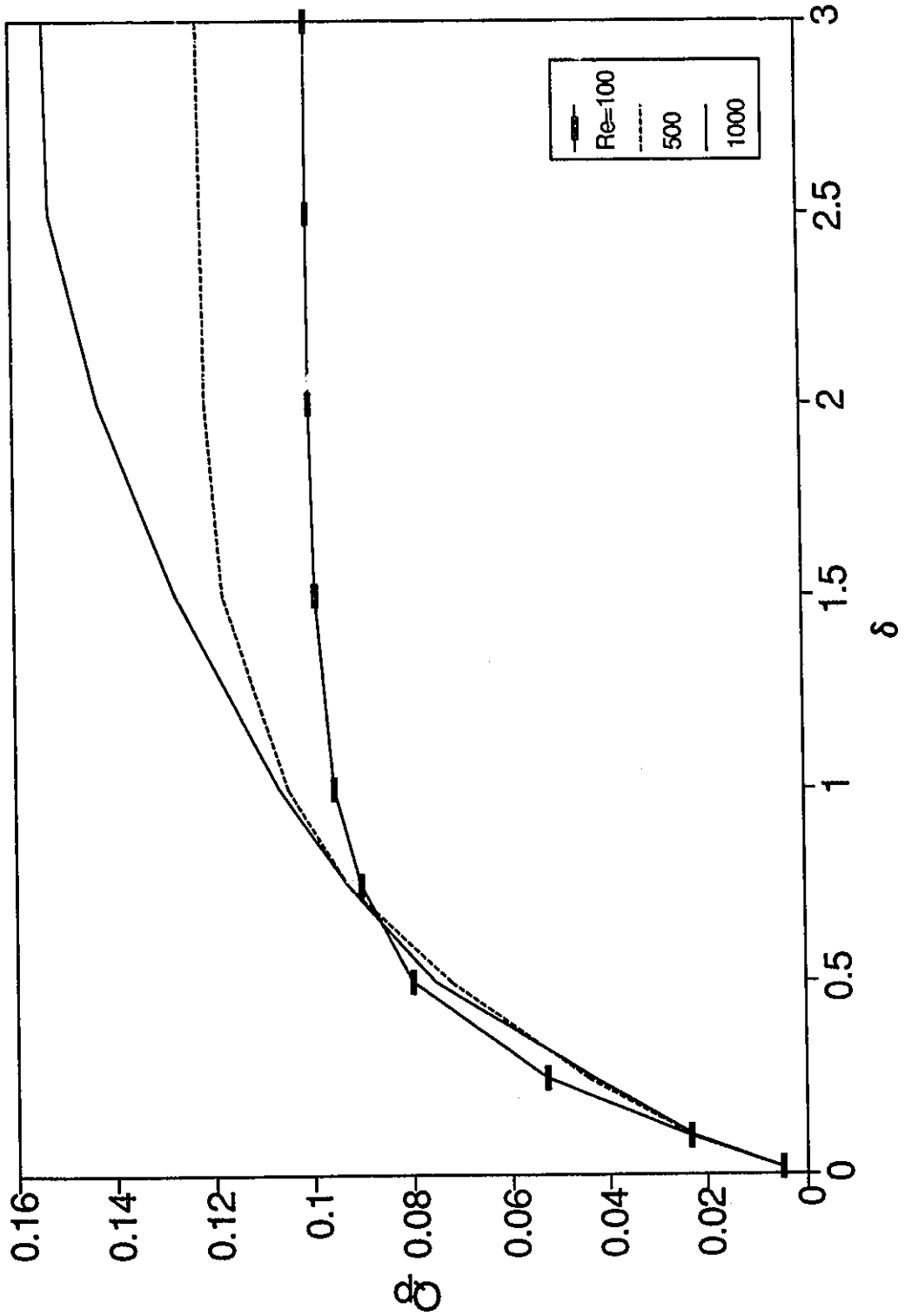

Figure 4.87 Primary Volumetric Flow Rate as a Function of δ for b = 0.025

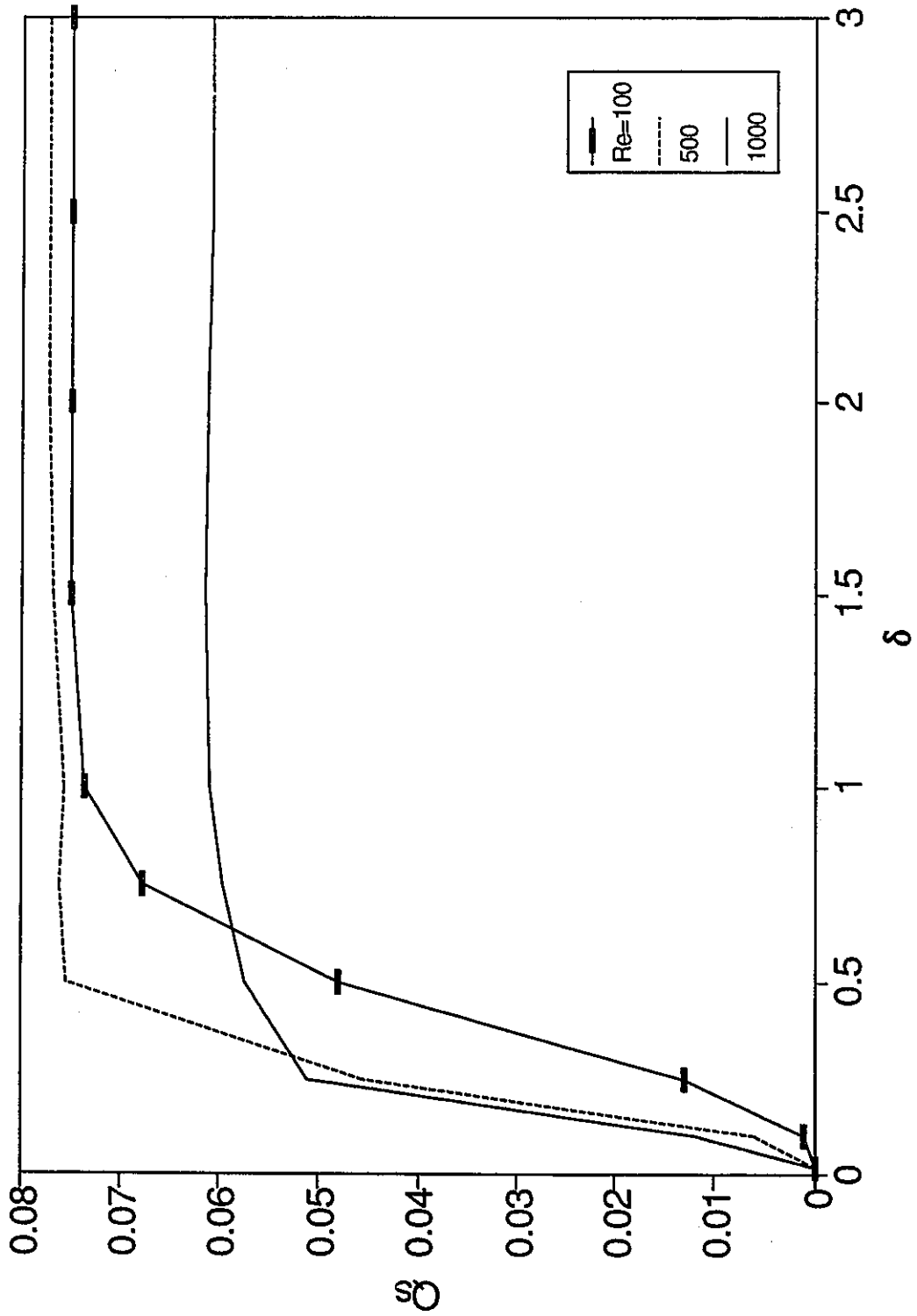

Figure 4.88 Secondary Volumetric Flow Rate as a Function of δ for b = 0.025

APPENDIX A: COEFFICIENTS FOR ADI SOLVERS

The following are the coefficients for equations (3.6):

$$A_{ij}^{*} = -\frac{\Delta t}{4\Delta x}u_{ij}^{n} - \frac{\Delta t}{2\Delta x^2 Re}, \tag{A.1}$$

$$B^{*} = \frac{\Delta t}{\Delta x^2 Re}, \tag{A.2}$$

$$C_{ij}^{*} = \frac{\Delta t}{4\Delta x}u_{ij}^{n} - \frac{\Delta t}{2\Delta x^2 Re}, \tag{A.3}$$

$$A_{ij} = -\frac{\Delta t}{4\Delta y}v_{ij}^{n} - \frac{\Delta t}{2\Delta y^2 Re}, \tag{A.4}$$

$$B = \frac{\Delta t}{\Delta y^2 Re}, \tag{A.5}$$

$$C_{ij} = \frac{\Delta t}{4\Delta y}v_{ij}^{n} - \frac{\Delta t}{2\Delta y^2 Re}, \tag{A.6}$$

where Δt is the time step.

The following are the coefficients for equations (3.20):

$$A_i^* = -\delta^2\left(\left.\frac{d\xi}{dr}\right|_i\right)^2 + \frac{\Delta\xi\delta^2}{2}\left[\left.\frac{d^2\xi}{dr^2}\right|_i - \frac{1}{r(\xi_i)}\left.\frac{d\xi}{dr}\right|_i\right], \tag{A.7}$$

$$B_i^* = 2\delta^2\left(\left.\frac{d\xi}{dr}\right|_i\right)^2, \tag{A.8}$$

$$C_i^* = -\delta^2\left(\left.\frac{d\xi}{dr}\right|_i\right)^2 - \frac{\Delta\xi\delta^2}{2}\left[\left.\frac{d^2\xi}{dr^2}\right|_i - \frac{1}{r(\xi_i)}\left.\frac{d\xi}{dr}\right|_i\right], \tag{A.9}$$

$$A_j = -\beta^2\left(\left.\frac{d\eta}{dz}\right|_j\right)^2 + \left.\frac{\Delta\xi\beta}{2}\frac{d^2\eta}{dz^2}\right|_j, \tag{A.10}$$

$$B_j = 2\beta^2\left(\left.\frac{d\eta}{dz}\right|_j\right)^2, \tag{A.11}$$

$$C_j = -\beta^2\left(\left.\frac{d\eta}{dz}\right|_j\right)^2 - \left.\frac{\Delta\xi\beta}{2}\frac{d^2\eta}{dz^2}\right|_j, \tag{A.12}$$

where $\beta = \Delta\xi/\Delta\eta$, $\sigma_\psi = \Delta\xi^2/\Delta t$ and Δt is the time step.

The following are the coefficients for equation (3.21):

$$A_{ij}^{*} = -\frac{\Delta\xi Re}{2}\frac{d\xi}{dr}\bigg|_i u_{i-1,j}^{n} - \left(\frac{d\xi}{dr}\bigg|_i\right)^2 + \frac{\Delta\xi}{2}\left[\frac{d^2\xi}{dr^2}\bigg|_i + \frac{1}{r(\xi_i)}\frac{d\xi}{dr}\bigg|_i\right], \tag{A.13}$$

$$B_i^{*} = 2\left(\frac{d^2\xi}{dr^2}\bigg|_i\right)^2 + \frac{\Delta\xi^2}{r(\xi_i)^2}, \tag{A.14}$$

$$C_{ij}^{*} = \frac{\Delta\xi Re}{2}\frac{d\xi}{dr}\bigg|_i u_{i+1,j}^{n} - \left(\frac{d\xi}{dr}\bigg|_i\right)^2 - \frac{\Delta\xi}{2}\left[\frac{d^2\xi}{dr^2}\bigg|_i + \frac{1}{r(\xi_i)}\frac{d\xi}{dr}\bigg|_i\right], \tag{A.15}$$

$$A_{ij} = -\frac{\Delta\xi\beta Re}{2}\frac{d\eta}{dz}\bigg|_j w_{i,j-1}^{n} - \frac{\beta^2}{\delta^2}\left(\frac{d\eta}{dz}\bigg|_j\right)^2 + \frac{\Delta\xi\beta}{2\delta^2}\frac{d^2\eta}{dz^2}\bigg|_j, \tag{A.16}$$

$$B_j = \frac{2\beta^2}{\delta^2}\left(\frac{d\eta}{dz}\bigg|_j\right)^2, \tag{A.17}$$

$$C_{ij} = \frac{\Delta\xi\beta Re}{2}\frac{d\eta}{dz}\bigg|_j w_{i,j+1}^{n} - \frac{\beta^2}{\delta^2}\left(\frac{d\eta}{dz}\bigg|_j\right)^2 - \frac{\Delta\xi\beta}{2\delta^2}\frac{d^2\eta}{dz^2}\bigg|_j, \tag{A.18}$$

$$E_j = \Delta\xi\beta Re\frac{d\eta}{dz}\bigg|_j, \tag{A.19}$$

where $\beta = \Delta\xi/\Delta\eta$, $\sigma_\zeta = \Delta\xi^2 Re/\Delta t$ and Δt is the time step.

The following are the coefficients for equations (3.22):

$$A_{ij}^{*} = -\frac{\Delta\xi Re}{2}\frac{d\xi}{dr}\bigg|_i u_{i-1,j}^{n} - \left(\frac{d\xi}{dr}\bigg|_i\right)^2 + \frac{\Delta\xi}{2}\left[\frac{d^2\xi}{dr^2}\bigg|_i + \frac{1}{r(\xi_i)}\frac{d\xi}{dr}\bigg|_i\right], \tag{A.20}$$

$$B_i^{*} = \frac{2\Delta\xi^2 Re}{r(\xi_i)}u_{ij}^{n} + 2\left(\frac{d\xi}{dr}\bigg|_i\right)^2 + \frac{\Delta\xi}{r(\xi_i)^2}, \tag{A.21}$$

$$C_{ij}^{*} = \frac{\Delta\xi Re}{2}\frac{d\xi}{dr}\bigg|_i u_{i+1,j}^{n} - \left(\frac{d\xi}{dr}\bigg|_i\right)^2 - \frac{\Delta\xi}{2}\left[\frac{d^2\xi}{dr^2}\bigg|_i + \frac{1}{r(\xi_i)}\frac{d\xi}{dr}\bigg|_i\right], \tag{A.22}$$

$$A_{ij} = -\frac{\Delta\xi\beta Re}{2}\frac{d\eta}{dz}\bigg|_j w_{ij-1}^{n} - \frac{\beta^2}{\delta^2}\left(\frac{d\eta}{dz}\bigg|_j\right)^2 + \frac{\Delta\xi\beta}{2\delta^2}\frac{d^2\eta}{dz^2}\bigg|_j, \tag{A.23}$$

$$B_j = \frac{2\beta^2}{\delta^2}\left(\frac{d\eta}{dz}\bigg|_j\right)^2, \tag{A.24}$$

$$C_{ij} = \frac{\Delta\xi\beta Re}{2}\frac{d\eta}{dz}\bigg|_j w_{ij+1}^{n} - \frac{\beta^2}{\delta^2}\left(\frac{d\eta}{dz}\bigg|_j\right)^2 - \frac{\Delta\xi\beta}{2\delta^2}\frac{d^2\eta}{dz^2}\bigg|_j, \tag{A.25}$$

where $\beta = \Delta\xi/\Delta\eta$, $\sigma_v = \Delta\xi^2 Re/\Delta t$ and Δt is the time step.

APPENDIX B: COMPUTER PROGRAMS FOR CAVITY FLOW

```
      PROGRAM CP
      PARAMETER ( MX = 129, MY = MX )
      PARAMETER ( MT = 25, MW = 11 )
*************************************************************************
*                                                                       *
*   PARAMETER SCAN OF CAVITY PROBLEM                                    *
*                                                                       *
*************************************************************************
      CHARACTER DFNAME*6, DFNAM2*6
      INTEGER IC, ICA, ICLMT, ICN, IFDATA, IFIRST, II, IOF, IPNDMP, IT,
     A        ITCDD0,ITCDLT, ITCDOC, ITCDOF, ITCDUF, ITCODE, ITIME, ITOP,
     A         ITOPFG, ITOPST, IW, JCOUNT, JJ, JLIMIT, MAXIT, MINIC,
     A         MINIT, NT, NW, NX, NY
      DOUBLE PRECISION CONV, DT, PSI, RE, U, V, W, ZETA
      REAL DDT1, DTA, DTI, DW1, TIME, TOTCPU, WA, WI, WLB, WUB, WTEMP,
     A     DDT(MT), DW(MW), WSTORE(MW)

      COMMON /BLOCDP/ DDT1, DW1
      COMMON /BLOCD1/ DFNAME
      COMMON /BLOCD2/ TOTCPU, IPNDMP
      COMMON /BLOCKC/ CONV, IC, ICLMT, ITCDD0, ITCDLT, ITCDOC, ITCDOF,
     A                ITCDUF
      COMMON /BLOCKN/ NX, NY
      COMMON /BLOCKP/ DT, RE, W
      COMMON /BLOCKR/ DTA(MT), WA(MT,MW), TIME(MT,MW), ICA(MT,MW),
     A                ICN(MT), IOF(MT), NT, NW
      COMMON /BLOCKV/ PSI(MX,MY), U(MX,MY), V(MX,MY), ZETA(MX,MY,2)

      EXTERNAL DIVCHK, OVFLOW, UNFLOW


C-- SET NUMBER OF ALLOWED ERRORS TO ONE ---------------------------------
      CALL ERRNU2
      CALL ERRSET(207,0,-1,0,OVFLOW,1)
      CALL ERRSET(208,0,-1,0,UNFLOW,1)
      CALL ERRSET(209,0,-1,0,DIVCHK,1)
C-----------------------------------------------------------------------

C-- INPUT DATA -----------------------------------------------------------
      READ(5,*) DFNAME, RE, IFDATA, CONV
      READ(5,*) NX, NY
      READ(5,*) NT, NW, MINIC, ITOP, IT, IW
      READ(5,*) DTI, WI
      READ(5,*) DDT1, DW1
      READ(5,*) IPNDMP
      READ(5,*) DFNAM2, IFINPT
      READ(5,*) IKEYBD, ISWOSC
C-----------------------------------------------------------------------

C-- CHECK IF OUTPUT FILE ALREADY EXISTS ----------------------------------
      CALL CMS(IRCODE,'STATE FILE '//DFNAME)
      IF(IRCODE.EQ.0)THEN
        WRITE(6,*)'WARNING:  OUTPUT FILE ALREADY EXISTS'
        WRITE(13,*)'WARNING:  OUTPUT FILE ALREADY EXISTS'
*       STOP
      ENDIF
```

```
C-----------------------------------------------------------------------

C-- INITIAL DATA ------------------------------------------------------
      ITCDD0 = 20 900
      ITCDLT = 20 100
      ITCDOC = 20 600
      ITCDOF = 20 700
      ITCDUF = 20 800
      ITCODE = 20 000
      JLIMIT = 50
      MINIT = 3*MINIC
      MAXIT = 7*MINIC
      ICLMT = MAXIT

      TOTCPU = 0.
C-----------------------------------------------------------------------

C-- WRITE HEADING ------------------------------------------------------
      II = NX/2 + 1
      JJ = NY/2 + 1
      WRITE(6,500) DFNAME, NX, NY, RE, CONV, ICLMT, NT, NW,
     A             II, JJ, II, JJ
C-----------------------------------------------------------------------

C--  GET INPUT DATA FROM KEYBOARD ---------------------------------------
    5 IF(IKEYBD.EQ.1)THEN
        READ(12,*) NW, DTI, WI, DW1
        IF(NW.EQ.99)GOTO 125
      ENDIF
C-----------------------------------------------------------------------

C-- DEFINE INCREMENTS FOR DTA AND WA -----------------------------------
      CALL DELPAR ( DDT, DW, NT, NW )
C-----------------------------------------------------------------------

C-- SET TIME STEPS -----------------------------------------------------
      DTA(1) = DTI
      IF ( NT.GT.1 ) THEN
        DO 10 I = 2,NT
   10     DTA(I) = DTA(I-1) + DDT(I)
      ENDIF
C-----------------------------------------------------------------------

C-- SET RELAXATION PARAMETERS ------------------------------------------
      WA(1,1) = WI
      IF ( NW.GT.1 ) THEN
        DO 20 J = 2,NW
   20     WA(1,J) = WA(1,J-1) + DW(J)
        IF( ITOP.EQ.1 ) WA(1,NW) = WA(1,NW-1)
      ENDIF

      IF ( (NW.GT.1 .AND. NT.GT.1) .OR. (NT.GT.1) ) THEN
        DO 30 I = 2,NT
          DO 30 J = 1,NW
   30       WA(I,J) = WA(1,J)
      ENDIF
C-----------------------------------------------------------------------

C-- INITIALIZE MISCELLANEOUS ARRAYS ------------------------------------
      DO 40 J = 1,NW
```

```
   40   WSTORE(J) = WA(1,J)

      DO 50 I = 1,NT
        DO 50 J = 1,NW
          ICA(I,J) = ITCODE
   50     TIME(I,J) = REAL ( ITCODE )

      DO 60 I = 1,NT
        ICN(I) = 0
   60   IOF(I) = 0
C-----------------------------------------------------------------------

C-- SET UP GRID ----------------------------------------------------------
      CALL GRID
C-----------------------------------------------------------------------



C-----------------------------------------------------------------------
C                        START PARAMETER SCAN
C-----------------------------------------------------------------------

C-- CHOOSE PARAMETERS ----------------------------------------------------
      DO 120 I = IT,NT
        WRITE(6,*)
        IFGSTR = 0
        IFIRST = 1
        ITOPST = 0
        ITOPFG = ITOP
        J = IW
        JCOUNT = 0
        DT = DBLE( DTA(I) )
        WTEMP = 0.
   70   IF ( J.LE.NW ) THEN
   80     W =  DBLE( WA(I,J) )
C-----------------------------------------------------------------------

C------ SET UP BOUNDARY AND INITIAL CONDITIONS OR INITIALIZE THE -------
C       FUNCTION VALUES FROM A DATA FILE
          IF ( IFINPT.EQ.1 ) THEN
            CALL BCIC
          ELSEIF ( IFINPT.EQ.2 ) THEN
            CALL ICF(DFNAM2)
          ENDIF
C-----------------------------------------------------------------------

C------ SOLVE SYSTEM ---------------------------------------------------
          CALL TINIT
          CALL SOLVE(ISWOSC)
          CALL TUSED(ITIME)
C-----------------------------------------------------------------------

C------ RECORD ITERATIONS AND CPU TIME ( ITIME IS IN MILLISECONDS ) ----
          ICA(I,J) = IC
          TIME(I,J) = REAL(ITIME)/1000./60.
          TOTCPU = TOTCPU + TIME(I,J)
C-----------------------------------------------------------------------

C------ OUTPUT PARAMETER DATA ------------------------------------------
          WRITE(6,501) DT, W, IC, TIME(I,J), PSI(II,JJ), ZETA(II,JJ,1)
```

```
          WRITE(13,501) DT, W, IC, TIME(I,J), PSI(II,JJ), ZETA(II,JJ,1)
          IF ( IC.GE.ITCODE ) TIME(I,J) = REAL(IC)
C-------------------------------------------------------------------------



C------ LOCATE A PROPER DATA POINT AT OVERFLOW BOUNDARY ----------------
          IF ( ITOPFG.EQ.1 ) THEN

C-------- FLAG TO INDICATE CONVERGENT REGION HAS BEEN ENTERED ----------
            IF ( IC.LE.MAXIT ) ITOPST = 1
C-------------------------------------------------------------------------

C-------- FIRST TIME THROUGH WHEN ITERATION LIMIT HAS BEEN EXCEEDED ----
              IF( IFIRST.EQ.1 .AND. ITOPST.EQ.1 .AND. IC.GT.MAXIT ) THEN
                IF( ICA(I,J-1).GE.MINIT .AND. ICA(I,J-1).LE.MAXIT ) THEN
                   ITOPFG = 0
                   GOTO 100
                ENDIF
                WLB = WA(I,J-1)
                WUB = WA(I,J)
                DO 90 K = J,NW-1
   90             WA(I,K+1) = WSTORE(K)
                ICA(I,J+1) = ICA(I,J)
                IFIRST = 0
              ENDIF
C-------------------------------------------------------------------------

C-------- FOR TOO FEW OR MANY ITERATIONS STEP DOWN OR UP ---------------
              IF ( IFIRST.EQ.0 .AND. (IC.LT.MINIT.OR.IC.GT.MAXIT) ) THEN
                IF(IC.LT.ITCODE)THEN
                  WSTR = WA(I,J)
                  ICSTR = ICA(I,J)
                  TSTR = TIME(I,J)
                  IFGSTR = 1
                ENDIF
                IF ( IC.LT.MINIT ) WLB = SNGL(W)
                IF ( IC.GT.MAXIT ) WUB = SNGL(W)
                WTEMP = WA(I,J)
                WA(I,J) = (WLB + WUB)/2.
                JCOUNT = JCOUNT + 1
                IF ( JCOUNT.LE.JLIMIT .AND.
     A               ABS(WTEMP-WA(I,J)).GE.5.0E-4 ) GOTO 80

C---------- ILL-CONDITIONED POINT FOUND -------------------------------
                IF(IFGSTR.EQ.1)THEN
                  WA(I,J) = WSTR
                  ICA(I,J) = ICSTR
                  TIME(I,J) = TSTR
                ENDIF
                WRITE(6,503)
                ICN(I) = J
                IC = ICA(I,J)
                J = J + 1
                ITOPFG = 0
C-------------------------------------------------------------------------

C-------- NUMBER OF ITERATIONS IS IN THE PROPER RANGE ------------------
              ELSEIF( IFIRST.EQ.0 )THEN
                IC = ICA(I,J)
```

```
                J = J + 1
                ITOPFG = 0
              ENDIF
C-------------------------------------------------------------------------

          ENDIF
C-------------------------------------------------------------------------


C------ EXIT J-LOOP WHEN OVERFLOW POINT IS LOCATED ---------------------
  100     IF ( ICA(I,J).EQ.ITCDOF ) THEN
            IOF(I) = J
            GOTO 105
          ENDIF
C-------------------------------------------------------------------------

C------ GET A NEW SET OF PARAMETERS ------------------------------------
          J = J + 1
          GOTO 70
        ENDIF

C-------- SEND PARTIAL PARAMETER RESULT TO PUNCH ------------------------
  105   IF ( I.LT.NT.AND.IFDATA.EQ.2.AND.IPNDMP.EQ.2 ) THEN
          WRITE(6,502) TOTCPU
          CALL DCP
          DO 110 K = 1,4
  110      BACKSPACE(6)
        ENDIF
  120 CONTINUE
C-------------------------------------------------------------------------
      IF(IKEYBD.EQ.1)GOTO 5

C-------------------------------------------------------------------------
C                         END PARAMETER SCAN
C-------------------------------------------------------------------------


C-- OUTPUT PARAMETER DATA AND DATA FILES --------------------------------
  125 IF(IFDATA.EQ.1) CALL DCF
      WRITE(6,502) TOTCPU
      IF(IFDATA.EQ.2) THEN
        WRITE(6,504)
        DO 130 I = 1,NT
  130       WRITE(6,505) I, ICN(I), IOF(I)
        CALL DCP
      ENDIF
C-------------------------------------------------------------------------


C-- FORMAT STATEMENTS ---------------------------------------------------
  500 FORMAT(1X,'DATA FILE = FILE ',A6/1X,'GRID = ',I3,' X ',I3,4X,
     A   'RE = ',F6.1,4X,'CONV = ',1PE11.2,4X,'ICLMT = ',I6/
     A   1X,'NT = ',I4,3X,'NW = ',I4///4X,'DT',7X,
     A   'W',8X,'ITER',5X,'TIME(MIN)',3X,'PSI(',I3,',',I3,')',2X,
     A   'ZETA(',I3,',',I3,')'/ ' ',70('-') )
  501 FORMAT(1X,F7.4,1X,F8.5,5X,I5,4X,1PG8.2,1X,2(4X,E9.2))
  502 FORMAT(//1X,'TOTAL TIME = ',F7.2,1X,'MIN.'/)
  503 FORMAT(/' PROCEDURE TERMINATED: POINT IS ILL-CONDITIONED'//)
  504 FORMAT(/4X,'I',3X,'ICN',3X,'IOF'/' ',17('-')/)
```

```
  505 FORMAT(' ',3(I4,2X))
C-----------------------------------------------------------------------

      END
```

```
      SUBROUTINE DELPAR ( DDT, DW, NT, NW )
*************************************************************************
*                                                                       *
*    DEFINE INCREMENTS FOR DTA AND WA                                   *
*                                                                       *
*************************************************************************
      REAL DDT1, DW1, DDT(NT), DW(NW)

      COMMON /BLOCDP/ DDT1, DW1

C-- DEFINE FIRST ELEMENTS ( THESE ARE NOT USED IN THE MAIN PROGRAM ) ---
      DDT(1) = 0.
      DW(1) = 0.
C-----------------------------------------------------------------------

C-- DEFINE INCREMENTS FOR DTA ------------------------------------------
      IF ( NT.GE.2 ) THEN
        DO 10 I = 2,NT
   10     DDT(I) = DDT1
      ENDIF
C-----------------------------------------------------------------------

C-- DEFINE INCREMENTS FOR WA -------------------------------------------
      IF ( NW.GE.2 ) THEN
        DO 20 I = 2,NW
   20     DW(I) = DW1
      ENDIF
C-----------------------------------------------------------------------

      END
```

```
      SUBROUTINE GRID
      PARAMETER ( MX = 129, MY = MX )
*************************************************************************
*                                                                       *
*   SETUP COMPUTATIONAL GRID                                            *
*                                                                       *
*        NOTE: DO NOT COMPILE THIS SUBROUTINE USING OPT(3)              *
*                                                                       *
*************************************************************************
      INTEGER NX, NY
      DOUBLE PRECISION AX, AY, BX, BY, DX, DY, R, XP, Z, YP,
     A                 G1, G2, H1, H2
      COMMON /BLOCKG/ DX, DY, R(MX), Z(MY),G1(MX),G2(MX),H1(MY),H2(MY)
      COMMON /BLOCKN/ NX, NY
      COMMON /BLOCKS/ AX,AY,BX,BY,ISR,ISZ

C-- INITIAL DATA ----------------------------------------------------------
      XP = 0.
      YP = 0.
      DX = 1./DFLOAT(NX - 1)
      DY = 1./DFLOAT(NY - 1)
C-----------------------------------------------------------------------

C-- CALCULATE GRID COORDINATES ---------------------------------------------
      DO 10 I = 2,NX
        XP = XP + DX
        IF(ISR.EQ.0)THEN
          R(I) = XP
        ELSE
          R(I) = ((BX+2.*AX)*((BX+1.)/(BX-1.))**((XP-AX)/(1.-AX))
     A          -BX+2.*AX)/((2.*AX+1.)*(1.+((BX+1.)/(BX-1))**
     A          ((XP-AX)/(1.-AX))))
        ENDIF
   10 CONTINUE
      DO 20 J = 2,NY
        YP = YP + DY
        IF(ISZ.EQ.0)THEN
          Z(J) = YP
        ELSE
          Z(J) = ((BY+2.*AY)*((BY+1.)/(BY-1.))**((YP-AY)/(1.-AY))
     A          -BY+2.*AY)/((2.*AY+1.)*(1.+((BY+1.)/(BY-1))**
     A          ((YP-AY)/(1.-AY))))
        ENDIF
   20 CONTINUE
C-----------------------------------------------------------------------

C--  CALCULATE METRICS ----------------------------------------------------
      DO 30 I = 1,NX
        IF(ISR.EQ.0)THEN
          G1(I) = 1.
          G2(I) = 0.
        ELSE
          G1(I) = 2.*BX*(1.-AX)*(2.*AX+1.)/DLOG((BX+1)/(BX-1))/
     A            (BX**2-((2.*AX+1.)*R(I)-2.*AX)**2)
          G2(I) = 4.*BX*(1.-AX)*(2.*AX+1.)**2*((2.*AX+1.)*R(I)-2.*AX)/
     A       DLOG((BX+1.)/(BX-1.))/(BX**2-((2.*AX+1.)*R(I)-2.*AX)**2)**2
        ENDIF
   30 CONTINUE
      DO 40 J = 1,NY
        IF(ISZ.EQ.0)THEN
```

```
          H1(J) = 1.
          H2(J) = 0.
        ELSE
          H1(J) = 2.*BY*(1.-AY)*(2.*AY+1.)/DLOG((BY+1)/(BY-1))/
     A            (BY**2-((2.*AY+1.)*Z(J)-2.*AY)**2)
          H2(J) = 4.*BY*(1.-AY)*(2.*AY+1.)**2*((2.*AY+1.)*Z(J)-2.*AY)/
     A       DLOG((BY+1.)/(BY-1.))/(BY**2-((2.*AY+1.)*Z(J)-2.*AY)**2)**2
        ENDIF
   40 CONTINUE
C-----------------------------------------------------------------------

      END
```

```
      SUBROUTINE BCIC
      PARAMETER ( MX = 129, MY = MX )
*************************************************************************
*                                                                       *
*   SETUP BOUNDARY AND INITIAL CONDITIONS                               *
*                                                                       *
*************************************************************************
      INTEGER NX, NY
      DOUBLE PRECISION PSI, U, V, ZETA
      COMMON /BLOCKN/ NX, NY
      COMMON /BLOCKV/ PSI(MX,MY), U(MX,MY), V(MX,MY), ZETA(MX,MY,2)

C-- INITIAL CONDITIONS ----------------------------------------------------
      DO 10 I = 1,NX
        DO 10 J = 1,NY
          PSI(I,J) = 0.
          ZETA(I,J,1) = 0.
          ZETA(I,J,2) = 0.
          U(I,J) = 0.
   10     V(I,J) = 0.
C-----------------------------------------------------------------------

C-- BOUNDARY CONDITIONS FOR TOP -------------------------------------------
      DO 20 I = 1,NX
   20   U(I,NY) = 1.
C-----------------------------------------------------------------------

      END
```

```
      SUBROUTINE ICF(DFNAM2)
      PARAMETER ( MX = 129, MY = MX )
*************************************************************************
*                                                                       *
*  READ IN FUNCTIONS FROM A DATA FILE                                   *
*                                                                       *
*************************************************************************
      CHARACTER DFNAME*6, DFNAM2*6, NAME*4
      INTEGER NX, NY
      DOUBLE PRECISION PSI, U, V, ZETA
      REAL SCONV, SDT, SPSI(MX,MY), SRE, SU(MX,MY), SV(MX,MY), SW,
     A     SX(MX), SY(MY), SZETA(MX,MY)

      COMMON /BLOCKV/ PSI(MX,MY), U(MX,MY), V(MX,MY), ZETA(MX,MY,2)


C-- WRITE VARIABLE DATA FILE ------------------------------------------
      OPEN(1,FILE=DFNAM2,FORM='FORMATTED',STATUS='OLD')
      READ(1,*) DFNAME
      READ(1,*) SRE,SCONV,CPU,SDT,SW,NX,NY,IC
      READ(1,*) NAME, ( SX(I), I = 1,NX )
      READ(1,*) NAME, ( SY(J), J = 1,NY )
      READ(1,*) NAME, ( (SPSI(I,J), I = 1,NX), J = 1,NY )
      READ(1,*) NAME, ( (SZETA(I,J), I = 1,NX), J = 1,NY )
      READ(1,*) NAME, ( (SU(I,J), I = 1,NX), J = 1,NY )
      READ(1,*) NAME, ( (SV(I,J), I = 1,NX), J = 1,NY )
      CLOSE ( 1 )
C-------------------------------------------------------------------------

C-- CONVERT FUNCTION DATA TO DOUBLE PRECISION --------------------------
      DO 10 I = 1,NX
        DO 10 J = 1,NY
          PSI(I,J) = DBLE( SPSI(I,J) )
          U(I,J) = DBLE( SU(I,J) )
          V(I,J) = DBLE( SV(I,J) )
   10     ZETA(I,J,1) = DBLE( SZETA(I,J) )
C-------------------------------------------------------------------------

      END
```

```
      SUBROUTINE DCP
      PARAMETER ( MT =  130, MW = 11 )
*************************************************************************
*                                                                       *
*  CREATE DATA FILE FOR PARAMETER DATA                                  *
*                                                                       *
*************************************************************************
      CHARACTER DFNAME*6
      INTEGER IC, ICLMT, ICN, IOF, IPNDMP, ITCDD0, ITCDLT, ITCDOC,
     A        ITCDOF, ITCDUF, NT, NW, ICA
      DOUBLE PRECISION CONV, DT, RE, W
      REAL TIME, DTA, TOTCPU, WA, RICA(MT,MW)

      COMMON /BLOCD1/ DFNAME
      COMMON /BLOCD2/ TOTCPU, IPNDMP
      COMMON /BLOCKC/ CONV, IC, ICLMT, ITCDD0, ITCDLT, ITCDOC, ITCDOF,
     A                ITCDUF
      COMMON /BLOCKN/ NX, NY
      COMMON /BLOCKP/ DT, RE, W
      COMMON /BLOCKR/ DTA(MT), WA(MT,MW), TIME(MT,MW), ICA(MT,MW),
     A                ICN(MT), IOF(MT), NT, NW


C-- CONVERT INTEGER ITERATIONS INTO REAL DATA --------------------------
      DO 10 I = 1,NT
        DO 10 J = 1,NW
   10     RICA(I,J) = REAL( ICA(I,J) )
C-------------------------------------------------------------------------

C-- WRITE PARAMETER DATA FILE -------------------------------------------
      OPEN(1,FILE=DFNAME,FORM='FORMATTED',STATUS='UNKNOWN')
      WRITE(1,*) '''',DFNAME,''''
      WRITE(1,*) SNGL(RE),SNGL(CONV),TOTCPU,NX,NY,ICLMT
      WRITE(1,*) NT, NW
      WRITE(1,*) '''ICN ''',' ',( ICN(I), I = 1,NT )
      WRITE(1,*) '''IOF ''',' ',( IOF(I), I = 1,NT )
      WRITE(1,*) '''DT  ''',' ',( DTA(I), I = 1,NT )
      WRITE(1,*) '''W   ''',' ',( (WA(I,J), I = 1,NT), J = 1,NW )
      WRITE(1,*) '''ITER''',' ',( (RICA(I,J), I = 1,NT), J = 1,NW )
      WRITE(1,*) '''TIME''',' ',( (TIME(I,J), I = 1,NT), J = 1,NW )
      CLOSE (1)
C-------------------------------------------------------------------------

C-- SEND PARTIAL RESULTS TO VIRTUAL PUNCH -------------------------------
      IF ( IPNDMP.EQ.1.OR.IPNDMP.EQ.2 ) THEN
        CALL CMS ( IRCODE,'DISK DUMP CP OUTPUT' )
        CALL CMS ( IRCODE,'DISK DUMP FILE '//DFNAME )
      ENDIF
C-------------------------------------------------------------------------

      END
```

```
      SUBROUTINE DCF
      PARAMETER ( MX = 129, MY = MX )
*************************************************************************
*                                                                       *
*  CREATE DATA FILE FOR FUNCTIONS                                       *
*                                                                       *
*************************************************************************
      CHARACTER DFNAME*6
      INTEGER IC, ICLMT, IPNDMP, ITCDD0, ITCDLT, ITCDOC, ITCDOF, ITCDUF
      DOUBLE PRECISION CONV, DT, DX, DY, PSI, RE, U, V, W, X, Y, ZETA
      REAL CPU, SPSI(MX,MY), SU(MX,MY), SV(MX,MY), SX(MX), SY(MY),
     A     SZETA(MX,MY)

      COMMON /BLOCD1/ DFNAME
      COMMON /BLOCD2/ CPU, IPNDMP
      COMMON /BLOCKC/ CONV, IC, ICLMT, ITCDD0, ITCDLT, ITCDOC, ITCDOF,
     A                ITCDUF
      COMMON /BLOCKG/ DX, DY, X(MX), Y(MY)
      COMMON /BLOCKN/ NX, NY
      COMMON /BLOCKP/ DT, RE, W
      COMMON /BLOCKV/ PSI(MX,MY), U(MX,MY), V(MX,MY), ZETA(MX,MY,2)

      IF (IC.GT.20000)GOTO 600
C-- CONVERT VARIABLE DATA TO SINGLE PRECISION ------------------------
      DO 10 I = 1,NX
        SX(I) = SNGL( X(I) )
        DO 10 J = 1,NY
          SY(J) = SNGL( Y(J) )
          SPSI(I,J) = SNGL( PSI(I,J) )
          SU(I,J) = SNGL( U(I,J) )
          SV(I,J) = SNGL( V(I,J) )
   10     SZETA(I,J) = SNGL( ZETA(I,J,1) )
C-----------------------------------------------------------------------

C-- WRITE VARIABLE DATA FILE -----------------------------------------
      OPEN(1,FILE=DFNAME,FORM='FORMATTED',STATUS='UNKNOWN')
      WRITE(1,*) '''',DFNAME,''''
      WRITE(1,*) SNGL(RE),SNGL(CONV),CPU,SNGL(DT),SNGL(W),NX,NY,IC
      WRITE(1,*) '''X    ''',' ',( SX(I), I = 1,NX )
      WRITE(1,*) '''Y    ''',' ',( SY(J), J = 1,NY )
      WRITE(1,*) '''PSI  ''',' ',( (SPSI(I,J), I = 1,NX), J = 1,NY )
      WRITE(1,*) '''ZETA ''',' ',( (SZETA(I,J), I = 1,NX), J = 1,NY )
      WRITE(1,*) '''U    ''',' ',( (SU(I,J), I = 1,NX), J = 1,NY )
      WRITE(1,*) '''V    ''',' ',( (SV(I,J), I = 1,NX), J = 1,NY )
      CLOSE ( 1 )
C-----------------------------------------------------------------------
600   CONTINUE
      END
```

```
      SUBROUTINE SOLVE(ISWOSC)
      PARAMETER ( MX = 129, MY = MX )
*************************************************************************
*                                                                       *
*   SOLVE STREAM FUNCTION VORTICITY EQUATIONS                           *
*                                                                       *
*************************************************************************
      INTEGER IC, ICLMT, ICOEFF, IFCONV, ITCDD0, ITCDLT, ITCDOC, ITCDOF,
     A        ITCDUF, NX, NY
      DOUBLE PRECISION CONV, DPSI1, DPSI2, DX, DY, DZETA, PSI, TPSI,
     A                 U, V, X, Y, ZETA

      COMMON /BLOCKC/ CONV, IC, ICLMT, ITCDD0, ITCDLT, ITCDOC, ITCDOF,
     A                ITCDUF
      COMMON /BLOCKG/ DX, DY, X(MX), Y(MY)
      COMMON /BLOCKN/ NX, NY
      COMMON /BLOCKV/ PSI(MX,MY), U(MX,MY), V(MX,MY), ZETA(MX,MY,2)

C-- INITIAL DATA ----------------------------------------------------------
      IC = 0
      ICOEFF = 0
      I1 = NX - 2
      J1 = NY/2
C-----------------------------------------------------------------------


C-- START ITERATION --------------------------------------------------------
   10 IC = IC + 1
        IFCONV = 0
C-----------------------------------------------------------------------

C---- SOLVE VORTICITY AT INTERIOR ---------------------------------------
        CALL ZETADI(ICOEFF,*207,*208,*209)
C-----------------------------------------------------------------------

C---- SOLVE STREAM FUNCTION AT INTERIOR ---------------------------------
        CALL PSISOR(*207,*208,*209)
C-----------------------------------------------------------------------

C---- CALCULATE U - AND V - VELOCITIES  AT INTERIOR ----------------------
        DO 20 I = 2,NX-1
          DO 20 J = 2,NY-1
            U(I,J) =   ( PSI(I,J+1) - PSI(I,J-1) )/(2.*DY)
   20       V(I,J) = - ( PSI(I+1,J) - PSI(I-1,J) )/(2.*DX)
C-----------------------------------------------------------------------

C---- CALCULATE VORTICITY AT BOUNDARIES ---------------------------------
        DO 30 I = 2,NX-1
          ZETA(I,1,2)  = 2.*(PSI(I,2)    -PSI(I,1)    )/DY**2
   30     ZETA(I,NY,2) = 2.*(PSI(I,NY-1)-PSI(I,NY)+DY)/DY**2
        DO 40 J = 2,NY-1
          ZETA(1,J,2)  = 2.*(PSI(2,J)    -PSI(1,J)    )/DX**2
   40     ZETA(NX,J,2) = 2.*(PSI(NX-1,J)-PSI(NX,J)    )/DX**2
C-----------------------------------------------------------------------

C---- CONVERGENCE TEST FOR ZETA -----------------------------------------
        DO 50 I = 1,NX
          DO 50 J = 1,NY
            IF ( IFCONV.EQ.0 ) THEN
              DZETA = DABS(ZETA(I,J,1) - ZETA(I,J,2))
```

```
              IF ( DZETA.GT.CONV ) IFCONV = 1
            ENDIF
   50       ZETA(I,J,1) = ZETA(I,J,2)
C-----------------------------------------------------------------------

C-- CHECK FOR OSCILLATING SOLUTION ---------------------------------------
      IF(ISWOSC.NE.0)THEN
        DPSI1 = PSI(I1,J1) - TPSI
        IF ( DPSI1.NE.0. .AND. DPSI2.NE.0. ) THEN
          IEXP1 = 3 - IDINT( DLOG10(DABS(DPSI1)) )
          IEXP2 = 3 - IDINT( DLOG10(DABS(DPSI2)) )
          IF(IEXP1.LE.IEXP2)THEN
            IEXP = IEXP1
          ELSE
            IEXP = IEXP2
          ENDIF
          IP1 =   IDINT( DPSI1*10.**IEXP )
          IP2 = - IDINT( DPSI2*10.**IEXP )
          IF(IP1.EQ.IP2)THEN
            IC = ITCDOC
            RETURN
          ENDIF
        ENDIF
        DPSI2 = DPSI1
        TPSI = PSI(I1,J1)
      ENDIF
C-----------------------------------------------------------------------

C---- DECIDE WHETHER TO DO ANOTHER ITERATION ---------------------------
        IF( IFCONV.EQ.1.AND.IC.LT.ICLMT ) GOTO 10
C-----------------------------------------------------------------------


C-- CALCULATE VORTICITY AT CORNERS ---------------------------------------
      ZETA(1,1,1)   = ( ZETA(1,2,1)      + ZETA(  2,1,1)   ) / 2.
      ZETA(NX,1,1)  = ( ZETA(NX,2,1)     + ZETA(NX-1,1,1)  ) / 2.
      ZETA(1,NY,1)  = ( ZETA(2,NY,1)     + ZETA(1,NY-1,1)  ) / 2.
      ZETA(NX,NY,1) = ( ZETA(NX,NY-1,1) + ZETA(NX-1,NY,1) ) / 2.
C-----------------------------------------------------------------------

C--- RETURNS FOR NORMAL TERMINATION OR PROGRAM INTERRUPT ---------------
      IF( IC.GE.ICLMT ) IC = ITCDLT
      RETURN
  207 IC = ITCDOF
      RETURN
  208 IC = ITCDUF
      RETURN
  209 IC = ITCDD0
C-----------------------------------------------------------------------

      END
```

```
      SUBROUTINE ZETADI(ICOEFF,*,*,*)
      PARAMETER ( MX = 129, MY = MX )
*************************************************************************
*                                                                       *
*   SOLVE VORTICITY USING ALTERNATING DIRECTION IMPLICIT (ADI)          *
*                                                                       *
*************************************************************************
      INTEGER ICOEFF, IFDIV0, IFOVER, IFUNDR, NX, NY
      DOUBLE PRECISION A1, C1, CX1, CX2, CX3, CX4, CY1, CY2, CY3,
     A     CY4, DT, DX, DY, PSI, RE, U, V, W, X, Y, ZETA
      DOUBLE PRECISION A(MY), AH(MX), B(MY), BH(MX), C(MY), CH(MX),
     A     F(MY), FH(MX), WK1(MX), WK2(MY), ZETAH(MX,MY)

      COMMON /BLOCKE/ IFDIV0, IFOVER, IFUNDR
      COMMON /BLOCKG/ DX, DY, X(MX), Y(MY)
      COMMON /BLOCKN/ NX, NY
      COMMON /BLOCKP/ DT, RE, W
      COMMON /BLOCKV/ PSI(MX,MY), U(MX,MY), V(MX,MY), ZETA(MX,MY,2)

C-- SET INITIAL DATA ONLY THE FIRST TIME THROUGH ----------------------
      IF ( ICOEFF.EQ.0 ) THEN
        ICOEFF = 1
        IFOVER = 1
        IFUNDR = 1
        CX1 = DT/(4.*DX)
        CX2 = DT/(2.*RE*DX**2)
        CX3 = 1. + 2.*CX2
        CX4 = 1. - 2.*CX2
        CY1 = DT/(4.*DY)
        CY2 = DT/(2.*RE*DY**2)
        CY3 = 1. + 2.*CY2
        CY4 = 1. - 2.*CY2
        DO 10 I = 2,NX-1
   10     BH(I) = CX3
        DO 20 J = 2,NY-1
   20     B(J) = CY3
      ENDIF
C-----------------------------------------------------------------------

C-- TRANSFER BOUNDARY ZETA TO TEMPORARY STORAGE -----------------------
      DO 30 I = 1,NX
        ZETAH(I,1) = ZETA(I,1,1)
   30   ZETAH(I,NY) = ZETA(I,NY,1)
      DO 40 J = 2,NY-1
        ZETAH(1,J) = ZETA(1,J,1)
   40   ZETAH(NX,J) = ZETA(NX,J,1)
C-----------------------------------------------------------------------


C-----------------------------------------------------------------------
C            FIRST TIME STEP -- SOLVE BY LINES
C-----------------------------------------------------------------------

C---- CALCULATE COEFFICIENTS AND RIGHT-HAND SIDE ----------------------
      DO 70 J = 2,NY-1
        DO 50 I = 2,NX-1
          AH(I) = - CX1*U(I,J) - CX2
          CH(I) =   CX1*U(I,J) - CX2
          A1    =   CY1*V(I,J) + CY2
          C1    = - CY1*V(I,J) + CY2
```

```
   50     FH(I) = A1*ZETA(I,J-1,1) + CY4*ZETA(I,J,1) + C1*ZETA(I,J+1,1)
        FH(2)    = FH(2)    -   AH(2)*ZETA(1,J,1)
        FH(NX-1) = FH(NX-1) - CH(NX-1)*ZETA(NX,J,1)
C-------------------------------------------------------------------------

C---- TRIDIAGONAL SOLVER ------------------------------------------------
        CALL TRIDBL(AH,BH,CH,FH,WK1,NX,2,NX-1)
C-------------------------------------------------------------------------

C---- TRANSFER SOLUTION TO TEMPORARY STORAGE-----------------------------
        DO 60 I = 2,NX-1
   60     ZETAH(I,J) = FH(I)
   70 CONTINUE
C-------------------------------------------------------------------------


C-------------------------------------------------------------------------
C             SECOND TIME STEP -- SOLVE BY LINES
C-------------------------------------------------------------------------

C---- CALCULATE COEFFICIENTS AND RIGHT-HAND SIDE ------------------------
       DO 100 I = 2,NX-1
         DO 80 J = 2,NY-1
           A(J) = - CY1*V(I,J) - CY2
           C(J) =   CY1*V(I,J) - CY2
           A1   =   CX1*U(I,J) + CX2
           C1   = - CX1*U(I,J) + CX2
   80      F(J) = A1*ZETAH(I-1,J) + CX4*ZETAH(I,J) + C1*ZETAH(I+1,J)
         F(2)    = F(2)    -   A(2)*ZETA(I,1,1)
         F(NY-1) = F(NY-1) - C(NY-1)*ZETA(I,NY,1)
C-------------------------------------------------------------------------

C---- TRIDIAGONAL SOLVER ------------------------------------------------
        CALL TRIDBL(A,B,C,F,WK2,NY,2,NY-1)
C-------------------------------------------------------------------------

C---- TRANSFER SOLUTION -------------------------------------------------
         DO 90 J = 2,NY-1
   90      ZETA(I,J,2) = F(J)
  100 CONTINUE
C-------------------------------------------------------------------------


C-- ALTERNATE RETURNS FOR PROGRAM INTERRUPTS --------------------------
       IF ( IFOVER.EQ.2 ) RETURN 1
       IF ( IFUNDR.EQ.2 ) RETURN 2
       IF ( IFDIV0.EQ.2 ) RETURN 3
C-------------------------------------------------------------------------

       END
```

```
      SUBROUTINE PSISOR(*,*,*)
      PARAMETER ( MX = 129, MY = MX )
*************************************************************************
*                                                                       *
*   SOLVE STREAM FUNCTION USING SUCCESSIVE OVERRELAXATION (SOR)         *
*                                                                       *
*************************************************************************
      INTEGER IFDIV0, IFOVER, IFUNDR, NX, NY
      DOUBLE PRECISION BT, DT, DX, DY, PSI, RE, U, V. W, X, Y, ZETA

      COMMON /BLOCKE/ IFDIV0, IFOVER, IFUNDR
      COMMON /BLOCKG/ DX, DY, X(MX), Y(MY)
      COMMON /BLOCKN/ NX, NY
      COMMON /BLOCKP/ DT, RE, W
      COMMON /BLOCKV/ PSI(MX,MY), U(MX,MY), V(MX,MY), ZETA(MX,MY,2)


C-- INITIAL DATA ----------------------------------------------------------
      IFOVER = 1
      IFUNDR = 1
      BT = DX/DY
C-----------------------------------------------------------------------

C-- SOLVE STREAM FUNCTION AT INTERIOR ---------------------------------
      DO 10 I = 2,NX-1
        DO 10 J = 2,NY-1
          PSI(I,J) = ( 1. - W )*PSI(I,J) + W*( PSI(I+1,J) + PSI(I-1,J)
     A         + BT**2*( PSI(I,J+1) + PSI(I,J-1) )
     A         - DX**2*ZETA(I,J,2) )/( 2.*( 1. + BT**2 ) )
   10 CONTINUE
C-----------------------------------------------------------------------

C-- ALTERNATE RETURNS FOR PROGRAM INTERRUPTS --------------------------
      IF ( IFOVER.EQ.2 ) RETURN 1
      IFUNDR = 1
      IF ( IFUNDR.EQ.2 ) RETURN 2
      IF ( IFDIV0.EQ.2 ) RETURN 3
C-----------------------------------------------------------------------

      END
```

```
      SUBROUTINE TRIDBL(A,B,C,F,X,N,NT,NB)
*********************************************************************
*                                                                   *
*    TRIDBL:  TRIDIAGONAL SOLVER USING DOUBLE PRECISION             *
*                                                                   *
*    KEY:                                                           *
*      A(*) = LOWER DIAGONAL                                        *
*      B(*) = DIAGONAL                                              *
*      C(*) = UPPER DIAGONAL                                        *
*      X(*) = WORKING ARRAY                                         *
*      F(*) = F ARRAY (CONTAINS SOLUTION AT EXIT FROM MODULE)       *
*      N = LENGTH OF ARRAY                                          *
*      NB = BOTTOM LIMIT FOR THE CALCULATION                        *
*      NT = TOP LIMIT FOR THE CALCULATION                           *
*                                                                   *
*********************************************************************
      INTEGER N,NB,NT
      DOUBLE PRECISION A(N),B(N),C(N),F(N),X(N),Z

C--- NORMALIZE FIRST ROW ---------------------------------------------
      X(NT) = C(NT)/B(NT)
      F(NT) = F(NT)/B(NT)
C---------------------------------------------------------------------

C--- REDUCE THE DIAGONAL TO 1'S --------------------------------------
      DO 10 J = NT+1,NB
         Z = 1./(B(J) - A(J)*X(J-1))
         X(J) = C(J)*Z
         F(J) = (F(J) - A(J)*F(J-1))*Z
   10 CONTINUE
C---------------------------------------------------------------------

C--- SWEEP BACKWARDS TO SOLVE PLACING THE SOLUTION IN F ---------------
      DO 20 J = NB-1,NT,-1
         F(J) = F(J) - X(J)*F(J+1)
   20 CONTINUE
C---------------------------------------------------------------------

      END
```

```
      SUBROUTINE OVFLOW
*****************************************************************************
*                                                                           *
*   SET OVERFLOW FLAG FOR ALTERNATE RETURN                                  *
*                                                                           *
*****************************************************************************
      COMMON /BLOCKE/ IFDIV0, IFOVER, IFUNDR
      INTEGER IFDIV0, IFOVER, IFUNDR

C-- SET OVERFLOW FLAG IF DIVIDE CHECK OR UNDERFLOW HAS NOT OCCURRED
      FIRST IF ( IFDIV0.NE.2 .AND. IFUNDR.NE.2 ) IFOVER = 2
C-----------------------------------------------------------------------

      END
      SUBROUTINE UNFLOW
*****************************************************************************
*                                                                           *
*   SET UNDERFLOW FLAG FOR ALTERNATE RETURN                                 *
*                                                                           *
*****************************************************************************
      COMMON /BLOCKE/ IFDIV0, IFOVER, IFUNDR
      INTEGER IFDIV0, IFOVER, IFUNDR

C-- SET UNDERFLOW FLAG IF OVERFLOW OR DIVIDE CHECK HAS NOT OCCURRED
      FIRST IF ( IFDIV0.NE.2 .AND. IFOVER.NE.2 ) IFUNDR = 2
C-----------------------------------------------------------------------

      END
      SUBROUTINE DIVCHK
*****************************************************************************
*                                                                           *
*   SET DIVIDE CHECK FLAG FOR ALTERNATE RETURN                              *
*                                                                           *
*****************************************************************************
      COMMON /BLOCKE/ IFDIV0, IFOVER, IFUNDR
      INTEGER IFDIV0, IFOVER, IFUNDR

C-- SET DIVIDE CHECK FLAG IF OVERFLOW OR UNDERFLOW HAS NOT OCCURRED
      FIRST IF ( IFOVER.NE.2 .AND. IFUNDR.NE.2 ) IFUNDR = 2
C-----------------------------------------------------------------------

      END
      SUBROUTINE ERRNU2
*****************************************************************************
*                                                                           *
*   DOES NOT INCLUDE OVERFLOW, UNDERFLOW OR DIVIDE CHECK                    *
*                                                                           *
*****************************************************************************

C-- SET ALLOWABLE NUMBER OF ERRORS TO ONE ------------------------------
      CALL ERRSET(152,1,0,0,0,152)
      CALL ERRSET(154,1,0,0,0,155)
      CALL ERRSET(159,1,0,0,0,161)
      CALL ERRSET(166,1,0,0,0,166)
      CALL ERRSET(169,1,0,0,0,189)
      CALL ERRSET(191,1,0,0,0,197)
      CALL ERRSET(199,1,0,0,0,201)
      CALL ERRSET(203,1,0,0,0,204)
      CALL ERRSET(206,1,0,0,0,206)
      CALL ERRSET(210,1,0,0,0,229)
```

```
      CALL ERRSET(231,1,0,0,0,239)
      CALL ERRSET(241,1,0,0,0,301)
C-----------------------------------------------------------------------

      END
```

APPENDIX C: COMPUTER PROGRAMS FOR ROTATING FLOW

```
      PROGRAM ROT
************************************************************************
*                                                                      *
*   ROTATING FLOW PROBLEM                                              *
*                                                                      *
************************************************************************
      INTEGER IDATAI, IR1, IR2, ITERM
      DOUBLE PRECISION CPU, CPU1, CPU2

      EXTERNAL INRUPT


C-- SET NUMBER OF ALLOWED ERRORS TO ONE --------------------------------
      CALL ERRNU2
      CALL ERRSET(207,2,-1,0,INRUPT,209)
C-----------------------------------------------------------------------

C-- READ INPUT DATA -----------------------------------------------------
      CALL ROTDAT(IDATAI,ITERM)
C-----------------------------------------------------------------------


C-- SET UP GRID ---------------------------------------------------------
      CALL ROTGRD
C-----------------------------------------------------------------------


C-- SET UP BOUNDARY AND INITIAL CONDITIONS OR INITIALIZE THE FUNCTION
C   VALUES FROM A DATA FILE ---------------------------------------------
      IF(IDATAI.NE.0)CALL ROTINP
      CALL ROTBIC(IDATAI)
C-----------------------------------------------------------------------

C-- SOLVE SYSTEM --------------------------------------------------------
      CALL CPUTIME(CPU1,IR1)
      CALL ROTSOL(ITERM)
      CALL CPUTIME(CPU2,IR2)
      CPU = (CPU2 - CPU1)*1.0D-6/60.
C-----------------------------------------------------------------------

C-- OUTPUT DATA FILES ---------------------------------------------------
      CALL ROTBLK
      CALL ROTOUT(CPU)
C-----------------------------------------------------------------------

      END
```

```
      SUBROUTINE ROTDAT(ISFNIN,ISTERM)
*****************************************************************
*                                                               *
*   READ INPUT DATA FOR ROTATING FLOW                           *
*                                                               *
*****************************************************************
      CHARACTER FNINP*6, FNOUT*6
      INTEGER IB, IC, ICMAX, IRCODE, ISFNIN, ISR, ISTERM, ISZ, NETA, NXI
      DOUBLE PRECISION AR, ALPHAR, ALPHAZ, B, BETAR, BETAZ, CONV,
     A  OMEGAB, OMEGAS, OMEGAT, RE, SIGZET

      COMMON /BLOCKA/ OMEGAB, OMEGAS, OMEGAT
      COMMON /BLOCKB/ IB
      COMMON /BLOCKC/ CONV, IC, ICMAX
      COMMON /BLOCKI/ FNINP
      COMMON /BLOCKN/ NXI, NETA
      COMMON /BLOCKO/ FNOUT
      COMMON /BLOCKP/ RE, AR, B, SIGZET
      COMMON /BLOCKS/ ALPHAR, ALPHAZ, BETAR, BETAZ, ISR, ISZ


C-- INITIAL DATA ----------------------------------------------------
      CONV = 1.0D-5
C-----------------------------------------------------------------------

C-- INPUT DATA -------------------------------------------------------
      OPEN(4,FILE='SIGDAT')
      OPEN(5)
      READ(5,*) NXI, NETA, ICMAX
      READ(5,*) RE, AR
      READ(5,*) ISR, ISZ, ALPHAR, ALPHAZ, BETAR, BETAZ
      READ(5,*) OMEGAB, OMEGAS, OMEGAT
      READ(5,*) FNOUT, ISFNIN, FNINP
      READ(5,*) ISTERM, IB
      READ(5,*) SIGZET
      IF(SIGZET.GE.500.) READ(4,*) SIGZET
      CLOSE(4)
      CLOSE(5)
C-----------------------------------------------------------------------

C-- CHECK IF OUTPUT FILE ALREADY EXISTS ------------------------------
      CALL CMS(IRCODE,'STATE FILE '//FNOUT)
      IRCODE = 1
      IF(IRCODE.EQ.0)THEN
        WRITE( 6,500) FNOUT
        WRITE(13,500) FNOUT
        STOP
      ENDIF
      WRITE(15,*) '''',FNOUT,''''
C-----------------------------------------------------------------------

C-- FORMAT STATEMENTS ------------------------------------------------
  500 FORMAT(' ','ERROR:  OUTPUT FILE (FILE ',A6,
     A       ') ALREADY EXISTS.  RUN TERMINATED.')
C-----------------------------------------------------------------------

      END
```

```
      SUBROUTINE ROTGRD
*     PARAMETER ( M = 101, N = M )
      INCLUDE (PARM)
*****************************************************************************
*                                                                           *
*   SETUP COMPUTATIONAL GRID FOR ROTATING FLOW                              *
*                                                                           *
*****************************************************************************
      INTEGER ISR, ISZ, NXI, NETA
      DOUBLE PRECISION A, ALPHAR, ALPHAZ, B, BETAR, BETAZ, DETA, DXI,
     A  D1CFNP, D1ETAZ, D1XIR, D2CFNP, D2ETAZ, D2XIR, ETA, F1, F2, F3,
     A  PFNC, R, VC, VP, XI, Z

      COMMON /BLOCKG/ DXI, DETA, R(M), Z(N), D1XIR(M), D2XIR(M),
     A  D1ETAZ(N), D2ETAZ(N)
      COMMON /BLOCKN/ NXI, NETA
      COMMON /BLOCKS/ ALPHAR, ALPHAZ, BETAR, BETAZ, ISR, ISZ

C--  STATEMENT FUNCTIONS ------------------------------------------------
      F1(VC,A,B) = ( (B+1.)/(B-1.) )**( (VC-A)/(1.-A) )
      F2(B) = DLOG( (B+1.)/(B-1.) )
      F3(VP,A,B) = B**2 - ((2.*A+1.)*VP-2.*A)**2

      PFNC(VC,A,B) = ( (2.*A+B)*F1(VC,A,B) - B + 2.*A )/
     A  ( (2.*A+1.)*(1.+F1(VC,A,B)) )

      D1CFNP(VP,A,B) = 2.*B*(1.-A)*(1.+2.*A)/F2(B)/F3(VP,A,B)
      D2CFNP(VP,A,B) = 4.*B*(1.-A)*(1.+2.*A)**2*( (1.+2.*A)*VP-2.*A )/
     A  F2(B)/F3(VP,A,B)**2
C-----------------------------------------------------------------------


C-- INITIAL DATA -------------------------------------------------------
      DXI  = 1./DFLOAT(NXI-1)
      DETA = 1./DFLOAT(NETA-1)
      XI  = - DXI
      ETA = - DETA
C-----------------------------------------------------------------------

C-- CALCULATE GRID COORDINATES AND METRICS -----------------------------
      DO 10 I = 1,NXI
        XI = XI + DXI
        IF(ISR.EQ.0)THEN
          R(I) = XI
          D1XIR(I) = 1.
          D2XIR(I) = 0.
          BETAR = 9.99
        ELSE
          R(I) = PFNC(XI,ALPHAR,BETAR)
          D1XIR(I) = D1CFNP(R(I),ALPHAR,BETAR)
          D2XIR(I) = D2CFNP(R(I),ALPHAR,BETAR)
        ENDIF
   10 CONTINUE
      R(1) = 0.

      DO 20 J = 1,NETA
        ETA = ETA + DETA
        IF(ISZ.EQ.0)THEN
          Z(J) = ETA
          D1ETAZ(J) = 1.
```

```
          D2ETAZ(J) = 0.
          BETAZ = 9.99
        ELSE
          Z(J) = PFNC(ETA,ALPHAZ,BETAZ)
          D1ETAZ(J) = D1CFNP(Z(J),ALPHAZ,BETAZ)
          D2ETAZ(J) = D2CFNP(Z(J),ALPHAZ,BETAZ)
        ENDIF
   20 CONTINUE
      Z(1) = 0.
C-----------------------------------------------------------------------

      END
```

```
      SUBROUTINE ROTINP
*     PARAMETER ( M = 101, N = M )
      INCLUDE (PARM)
*************************************************************************
*                                                                       *
*  READ IN FUNCTIONS FROM A DATA FILE  FOR ROTATING FLOW                *
*                                                                       *
*************************************************************************
      CHARACTER FNINP*6, FNOUT*6, NAME*4
      INTEGER IC, IR, IZ, NETA, NXI
      DOUBLE PRECISION PSI, U, V, W, ZETA
      REAL AR, ALPHAR, ALPHAZ, B, BETAR, BETAZ, CM, CONV, CPU, OMEGAB,
     A  OMEGAS, OMEGAT, QP, QS, RE, SIGZET
      REAL SR(M), SZ(N), SPSI(M,N), SU(M,N), SV(M,N), SW(M,N),
     A  SZETA(M,N)

      COMMON /BLOCKI/ FNOUT
      COMMON /BLOCKV/ PSI(M,N,2), U(M,N), V(M,N), W(M,N), ZETA(M,N)


C-- WRITE FUNCTION DATA FILE ------------------------------------------
      OPEN(1,FILE=FNOUT,FORM='FORMATTED',STATUS='OLD')
      READ(1,*) FNINP
      READ(1,*) RE,AR,B
      READ(1,*) NXI,NETA,SIGZET
      READ(1,*) IC,CPU,CONV
      READ(1,*) IR,IZ,ALPHAR,ALPHAZ,BETAR,BETAZ
      READ(1,*) OMEGAB,OMEGAS,OMEGAT
      READ(1,*) CM,QP, QS

      READ(1,*) NAME, ( SR(I), I = 1,NXI )
      READ(1,*) NAME, ( SZ(J), J = 1,NETA )

      READ(1,*) NAME, ( (SPSI(I,J), I = 1,NXI), J = 1,NETA )
      READ(1,*) NAME, ( (SZETA(I,J), I = 1,NXI), J = 1,NETA )
      READ(1,*) NAME, ( (SU(I,J), I = 1,NXI), J = 1,NETA )
      READ(1,*) NAME, ( (SV(I,J), I = 1,NXI), J = 1,NETA )
      READ(1,*) NAME, ( (SW(I,J), I = 1,NXI), J = 1,NETA )
      CLOSE ( 1 )

      WRITE(6,*) 'RE =',RE
C-------------------------------------------------------------------------

C-- CONVERT FUNCTION DATA TO DOUBLE PRECISION --------------------------
      DO 10 I = 1,NXI
        DO 10 J = 1,NETA
          PSI(I,J,1) = DBLE(SPSI(I,J))
          U(I,J) = DBLE(SU(I,J))
          V(I,J) = DBLE(SV(I,J))
          W(I,J) = DBLE(SW(I,J))
   10     ZETA(I,J) = DBLE(SZETA(I,J))
C-------------------------------------------------------------------------

      END
```

```
      SUBROUTINE ROTBIC(IDATAI)
*     PARAMETER ( M = 101, N = M )
      INCLUDE (PARM)
*************************************************************************
*                                                                       *
*   SETUP BOUNDARY AND INITIAL CONDITIONS  FOR ROTATING FLOW            *
*                                                                       *
*************************************************************************
      INTEGER IB, IDATAI, NETA, NXI
      DOUBLE PRECISION AR, B, DETA, DXI, OMEGAB, OMEGAS, OMEGAT, PSI, R,
     A  RE, U, V, W, Z, ZETA, DTEMP, SIGZET, VTEMP, TSIGZE

      COMMON /BLOCKA/ OMEGAB, OMEGAS, OMEGAT
      COMMON /BLOCKB/ IB
      COMMON /BLOCKG/ DXI, DETA, R(M), Z(N)
      COMMON /BLOCKN/ NXI, NETA
      COMMON /BLOCKP/ RE, AR, B, SIGZET
      COMMON /BLOCKV/ PSI(M,N,2), U(M,N), V(M,N), W(M,N), ZETA(M,N)

C-- INITIAL CONDITIONS ----------------------------------------------------
      IF(IDATAI.EQ.0)THEN
        DO 10 I = 1,NXI
          DO 10 J = 1,NETA
            PSI(I,J,1) = 0.
            PSI(I,J,2) = 0.
            ZETA(I,J) = 0.
            U(I,J) = 0.
            V(I,J) = 0.
   10       W(I,J) = 0.
      ENDIF
C-----------------------------------------------------------------------

C-- BOUNDARY CONDITIONS FOR V-VELOCITY -------------------------------------
      DO 20 I = 1,NXI
        V(I,1)    = OMEGAB*R(I)
        V(I,NETA) = OMEGAT*R(I)
   20 CONTINUE
      DO 30 J = 1,NETA
   30   V(NXI,J)  = OMEGAS
C-----------------------------------------------------------------------

C-- SET THE GAP LENGTH -----------------------------------------------------
      IF(IB.EQ.NXI)THEN
        B = 0.0D0
      ELSE
        B = 1.0D0 - R(IB)
      ENDIF
C-----------------------------------------------------------------------

C-- BOUNDARY CONDITION FOR COUETTE FLOW ------------------------------------
        IF(IB.LT.NXI)THEN
          DO 40 I = IB,NXI-1
            V(I,NETA) = ( R(I)*(OMEGAS-OMEGAT*(1.-B)**2)
     A          - (1.-B)**2*(OMEGAS-OMEGAT)/R(I) )/(2.*B-B**2)
   40     CONTINUE
        ENDIF
C-----------------------------------------------------------------------

C-- SOLVE V-VELOCITY FUNCTION AT INTERIOR ----------------------------------
      IF(IDATAI.EQ.0)THEN
```

```
        TSIGZE = SIGZET
        SIGZET = 1.0D0
        ISWTV = 1
        IVEL = 0
   50     IVEL = IVEL + 1
          VTEMP = V(NXI/2+1,NETA/2+1)
          CALL ROTVEL(ISWTV)
          DTEMP = DABS( (V(NXI/2+1,NETA/2+1)-VTEMP)/V(NXI/2+1,NETA/2+1))
        IF(DTEMP.GT.0.010 .AND. IVEL.LT.500)GOTO 50
        WRITE(13,500)IVEL
        SIGZET = TSIGZE
      ENDIF
C-----------------------------------------------------------------------

C-- FORMAT STATEMENTS ---------------------------------------------------
  500 FORMAT(' ',I5)
C-----------------------------------------------------------------------

      END
```

```
      SUBROUTINE ROTBLK
*     PARAMETER ( M = 101, N = M )
      INCLUDE (PARM)
************************************************************************
*                                                                      *
*   CALCULATE BULK QUANTITIES FOR ROTATING FLOW                        *
*                                                                      *
************************************************************************
      INTEGER IB, NETA, NXI
      DOUBLE PRECISION AR, B, CM, DETA, DXI, D1ETAZ, D1VETA, D1XIR,
     A  D2ETAZ,D2XIR, PI, PSI, PSIMAX, QP, QS, R, RE, SIGZET, U, V, W,
     A  Z, ZETA, A(N), UI(N), VI(M,N)

      COMMON /BLOCKB/ IB
      COMMON /BLOCKG/ DXI, DETA, R(M), Z(N), D1XIR(M), D2XIR(M),
     A  D1ETAZ(N), D2ETAZ(N)
      COMMON /BLOCKN/ NXI, NETA
      COMMON /BLOCKP/ RE, AR, B, SIGZET
      COMMON /BLOCKV/ PSI(M,N,2), U(M,N), V(M,N), W(M,N), ZETA(M,N)
      COMMON /BLOCKX/ CM, QP, QS, PSIMAX, D1VETA(M), IMAX, JMAX


C-- INITIAL DATA ----------------------------------------------------------
      PI = 3.14159 26535 89793 23846D0
C-----------------------------------------------------------------------

C-- FIND MAXIMUM VALUE OF THE STREAM FUNCTION ------------------------------
      CALL  MAXABS(PSI,M,N,NXI,NETA,PSIMAX,IMAX,JMAX)
C-----------------------------------------------------------------------

C-- CALCULATE PRIMARY VOLUMETRIC FLOW RATE ---------------------------------
      DO 10 I = 1,NXI
        DO 10 J = 1,NETA
   10     VI(I,J) = AR*V(I,J)/D1XIR(I)/D1ETAZ(J)

      CALL D2INTE(VI,M,N,NXI,NETA,DXI,DETA,QP)
C-----------------------------------------------------------------------

C-- CALCULATE SECONDARY VOLUMETRIC FLOW RATE -------------------------------
      DO 20 J = 1,NETA
   20   UI(J) = 2.*AR*PI*R(IMAX)*U(IMAX,J)/D1ETAZ(J)

      CALL SIMSON(UI,JMAX,DETA,QS)
C-----------------------------------------------------------------------

C-- CALCULATE TORQUE COEFFICIENT -------------------------------------------
      DO 30 I = 1,NXI-1
        D1VETA(I) = D1ETAZ(NETA)*(3.*V(I,NETA) - 4.*V(I,NETA-1)
     A    + V(I,NETA-2))/2./DETA
        A(I) = 8.*PI*D1VETA(I)*R(I)**2/D1XIR(I)/AR/RE
   30 CONTINUE

C---- CONDITION FOR PAO'S METHOD
      A(NXI  ) = 8.*PI*D1VETA(NXI-1)/R(NXI-1)*R(NXI  )**3/AR/RE

      CALL SIMSON(A,IB,DXI,CM)
C-----------------------------------------------------------------------

      END
```

```
      SUBROUTINE ROTOUT(CPU)
*     PARAMETER ( M = 101, N = M )
      INCLUDE (PARM)
************************************************************************
*                                                                      *
*  WRITE DATA FILES FOR ROTATING FLOW FUNCTIONS                        *
*                                                                      *
************************************************************************
      CHARACTER FNOUT*6
      INTEGER IC, ICMAX, II, JJ, ISR, ISZ, NETA, NXI
      DOUBLE PRECISION AR, ALPHAR, ALPHAZ, B, BETAR, BETAZ, CM, CONV,
     A  CPU, DETA, DXI, D1VETA, OMEGAB, OMEGAS, OMEGAT, PSI, PSIMAX,
     A  QP, QS, R, RE, SIGZET, U, V, W, Z, ZETA

      COMMON /BLOCKA/ OMEGAB, OMEGAS, OMEGAT
      COMMON /BLOCKC/ CONV, IC, ICMAX
      COMMON /BLOCKG/ DXI, DETA, R(M), Z(N)
      COMMON /BLOCKN/ NXI, NETA
      COMMON /BLOCKO/ FNOUT
      COMMON /BLOCKP/ RE, AR, B, SIGZET
      COMMON /BLOCKS/ ALPHAR, ALPHAZ, BETAR, BETAZ, ISR, ISZ
      COMMON /BLOCKV/ PSI(M,N,2), U(M,N), V(M,N), W(M,N), ZETA(M,N)
      COMMON /BLOCKX/ CM, QP, QS, PSIMAX, D1VETA(M), IMAX, JMAX


C-- CALCULATE DATA ----------------------------------------------------
      II = NXI/2 + 1
      JJ = NETA/2 + 1
C-----------------------------------------------------------------------

C-- WRITE FUNCTION DATA FILE -------------------------------------------
      OPEN(1,FILE=FNOUT,FORM='FORMATTED',STATUS='UNKNOWN')
      WRITE(1,*) '''', FNOUT, ''''
      WRITE(1,*) SNGL(RE),SNGL(AR),SNGL(B)
      WRITE(1,*) NXI,NETA,SNGL(SIGZET)
      WRITE(1,*) IC,SNGL(CPU),SNGL(CONV)
      WRITE(1,*) ISR,ISZ,SNGL(ALPHAR),SNGL(ALPHAZ),SNGL(BETAR),
     A           SNGL(BETAZ)
      WRITE(1,*) SNGL(OMEGAB),SNGL(OMEGAS),SNGL(OMEGAT)
      WRITE(1,*) SNGL(CM),SNGL(QP),SNGL(QS)

      WRITE(1,*) '''R   ''',' ',(SNGL(R(I)),I=1,NXI)
      WRITE(1,*) '''Z   ''',' ',(SNGL(Z(J)),J=1,NETA)

      WRITE(1,*) '''PSI ''',' ',((SNGL(PSI(I,J,1)),I=1,NXI),J=1,NETA)
      WRITE(1,*) '''ZETA''',' ',((SNGL(ZETA(I,J)),I=1,NXI),J=1,NETA)
      WRITE(1,*) '''U   ''',' ',((SNGL(U(I,J)),I=1,NXI),J= 1,NETA)
      WRITE(1,*) '''V   ''',' ',((SNGL(V(I,J)),I=1,NXI),J= 1,NETA)
      WRITE(1,*) '''W   ''',' ',((SNGL(W(I,J)),I=1,NXI),J= 1,NETA)
      CLOSE ( 1 )
C-----------------------------------------------------------------------

C-- WRITE HEADING AND PARAMETER DATA -----------------------------------
      WRITE(6,500) RE,AR,B
      WRITE(6,510) NXI,NETA,SIGZET
      WRITE(6,520) IC,CPU,CONV,FNOUT
      WRITE(6,530) ISR,ISZ,ALPHAR,ALPHAZ,BETAR,BETAZ
      WRITE(6,540) OMEGAB,OMEGAS,OMEGAT
      WRITE(6,550) II,JJ,PSI(II,JJ,1),II,JJ,ZETA(II,JJ)
      WRITE(6,560) IMAX, JMAX, PSIMAX
```

```
      WRITE(6,570) CM
      WRITE(6,580) QP
      WRITE(6,590) QS
      WRITE(6,600)
      DO 40 I = 1,NXI
        WRITE(6,610) I, R(I), V(I,NETA), D1VETA(I)
   40 CONTINUE

      WRITE(13,620) IC, CPU, PSI(II,JJ,1), ZETA(II,JJ)
C-----------------------------------------------------------------------

C-- FORMAT STATEMENTS ---------------------------------------------------
  500 FORMAT(1X,'CE = ',F7.0,4X,'AR = ',F5.3,4X,'B = ',F6.4/)
  510 FORMAT(1X,'GRID = ',I3,' X ',I3,4X,'SIGZET = ',F7.1/)
  520 FORMAT(1X,'ITERATIONS = ',I7,4X,'CPU = ',F6.2,4X,'CONV = ',E12.3,
     A  4X,'DATA FILE = ',A6/)
  530 FORMAT(1X,'ISR = ',I1,4X,'ISZ = ',I1,4X,'ALPHAR = ',F4.2,4X,
     A  'ALPHAR = ',F4.2,4X,'BETAR = ',F9.3,4X,'BETAZ = ',F8.3/)
  540 FORMAT(1X,'OMEGAB = ',F5.2,4X,'OMEGAS = ',F5.2,4X,'OMEGAT = ',
     A  F5.2//)
  550 FORMAT( '  PSI(',I3,',',I3,') = ',E9.2,/' ZETA(',I3,',',I3,') = ',
     A  E9.2)
  560 FORMAT(' ','PSIMAX(',I3,',',I3,') = ',E11.4)
  570 FORMAT(' ','CM = ',E11.4)
  580 FORMAT(' ','QP = ',E11.4)
  590 FORMAT(' ','QS = ',E11.4//)
  600 FORMAT('  I ',3X,'R(I)',6X,'V(I,TOP)',4X,'DV/DZ(I,TOP)'/)
  610 FORMAT(' ',I3,2X,F6.4,3(3X,E11.4))
  620 FORMAT(//I5,3X,1PG8.2,1X,2(3X,E9.2)//)
C-----------------------------------------------------------------------

      END
```

```
      SUBROUTINE ROTSOL(ITERM)
*     PARAMETER ( M = 101, N = M )
      INCLUDE (PARM)
************************************************************************
*                                                                      *
*   SOLVE ROTATING FLOW PROBLEM                                        *
*                                                                      *
************************************************************************
      INTEGER IC, ICMAX, ICODE, IFCONV, IRCODE, ISWTP, ISWTV, ISWTZ,
     A  ITERM, NXI, NETA
      DOUBLE PRECISION AR, B, CONV, DETA, DPSI, DXI, D1ETAZ, D2ETAZ,
     A  D1XIR, D2XIR, PSI, R, RE, SIGZET, U, V, W, Z, ZETA

      COMMON /BLOCKB/ IB
      COMMON /BLOCKC/ CONV, IC, ICMAX
      COMMON /BLOCKG/ DXI, DETA, R(M), Z(N), D1XIR(M), D2XIR(M),
     A  D1ETAZ(N), D2ETAZ(N)
      COMMON /BLOCKN/ NXI, NETA
      COMMON /BLOCKP/ RE, AR, B, SIGZET
      COMMON /BLOCKV/ PSI(M,N,2), U(M,N), V(M,N), W(M,N), ZETA(M,N)

C-- INITIAL DATA ----------------------------------------------------------
      IC = 0
      ISWTP = 1
      ISWTV = 1
      ISWTZ = 1
C-----------------------------------------------------------------------


C-- START ITERATION --------------------------------------------------------
   10 IC = IC + 1
        IFCONV = 0
C-----------------------------------------------------------------------

C---- SOLVE VORTICITY AT INTERIOR ---------------------------------------
        CALL ROTZET(ISWTZ)
C-----------------------------------------------------------------------

C---- SOLVE STREAM FUNCTION AT INTERIOR ---------------------------------
        CALL ROTPSI(ISWTP)
C-----------------------------------------------------------------------

C---- CALCULATE VORTICITY AT BOUNDARIES ---------------------------------
      DO 20 I = 2,NXI-1
        ZETA(I,1)    = 2.*(D1ETAZ(1)/DETA)**2/R(I)*PSI(I,2,2)
   20   ZETA(I,NETA) = 2.*(D1ETAZ(NETA)/DETA)**2/R(I)*PSI(I,NETA-1,2)
      DO 30 J = 2,NETA-1
   30   ZETA(NXI,J)  = 2.*(AR*D1XIR(NXI)/DXI)**2*PSI(NXI-1,J,2)
C-----------------------------------------------------------------------

C---- CALCULATE VORTICITY IN THE GAP FOR SYMMETRY CASE ------------------
*     DO 40 I = IB+1,NXI
*       ZETA(I,NETA) = 0.0D0
*  40 CONTINUE
C-----------------------------------------------------------------------

C---- SOLVE V-VELOCITY FUNCTION AT INTERIOR -----------------------------
        CALL ROTVEL(ISWTV)
C-----------------------------------------------------------------------
```

```
C---- CALCULATE U - AND W - VELOCITIES  AT INTERIOR --------------------
        DO 50 I = 2,NXI-1
          DO 50 J = 2,NETA-1
            U(I,J) =  D1ETAZ(J)*(PSI(I,J+1,2)-PSI(I,J-1,2))/2./R(I)/DETA
   50       W(I,J) = - D1XIR(I)*(PSI(I+1,J,2)-PSI(I-1,J,2))/2./R(I)/DXI
C-----------------------------------------------------------------------

C---- CALCULATE U - AND W - VEL. IN THE GAP FOR THE SYMMETRY CASE ------
*     DO 60 I = IB+1,NXI
*       V(I,NETA) = 4.D0/3.*V(I,NETA-1) - V(I,NETA-2)/3.
*       U(I,NETA) = 4.D0/3.*U(I,NETA-1) - U(I,NETA-2)/3.
*  60 CONTINUE
C-----------------------------------------------------------------------

C---- CALCULATE W-VELOCITY AT LINE OF SYMMETRY -------------------------
        DO 70 J = 2,NETA-1
   70     W(1,J) = W(2,J)
C-----------------------------------------------------------------------

C---- OUTPUT DATA TO TERMINAL AND DATA FILE ----------------------------
        IF(ITERM.EQ.1)THEN
        WRITE(15,500)IC,PSI(NXI/2+1,NETA/2+1,2),PSI(3*NXI/4,3*NETA/4,2),
     A       ZETA(NXI/2+1,NETA/2+1),ZETA(3*NXI/4,3*NETA/4)
        WRITE(13,500)IC,PSI(NXI/2+1,NETA/2+1,2),PSI(3*NXI/4,3*NETA/4,2),
     A       ZETA(NXI/2+1,NETA/2+1),ZETA(3*NXI/4,3*NETA/4)
        ENDIF
C-----------------------------------------------------------------------

C---- CONVERGENCE TEST FOR PSI -----------------------------------------
        DPSI = 0.0D0
        DO 80 I = 1,NXI
          DO 80 J = 1,NETA
            IF(PSI(I,J,1).NE.0.) DPSI =
     A        DABS( (PSI(I,J,1) - PSI(I,J,2))/PSI(I,J,1) )
   80       IF(DPSI.GT.CONV)GOTO 90
        IFCONV = 1
   90   CONTINUE
C-----------------------------------------------------------------------

C---- UPDATE PSI -------------------------------------------------------
        DO 100 I = 1,NXI
          DO 100 J = 1,NETA
  100       PSI(I,J,1) = PSI(I,J,2)
C-----------------------------------------------------------------------

C---- CHECK WHETHER A USER INTERRUPT IS REQUESTED -----------------------
        CALL CMS(IRCODE,'EXEC Q1F1')
        IF(IRCODE.EQ.0)THEN
          CALL CMS(IRCODE,'CP DETACH 1F1')
          WRITE(13,510) RE, SIGZET
          READ(12,*) ICODE
          IF(ICODE.EQ.1) GOTO 110
        ENDIF
C-----------------------------------------------------------------------

C---- DECIDE WHETHER TO DO ANOTHER ITERATION ---------------------------
        IF( IFCONV.EQ.0.AND.IC.LT.ICMAX ) GOTO 10
        IF(IC.LE.1) GOTO 10
C-----------------------------------------------------------------------
```

```
C-- CALCULATE VORTICITY AT CORNERS ------------------------------------
  110 ZETA(NXI,1) = ( ZETA(NXI,2) + ZETA(NXI-1,1) )/2.
      ZETA(NXI,NETA) = ( ZETA(NXI,NETA-1) + ZETA(NXI-1,NETA) )/2.
C-----------------------------------------------------------------------

C--- SET IC IF LIMIT ICMAX IS REACHED ---------------------------------
      IF( IC.GE.ICMAX )THEN
        IC = 20100
        OPEN(9,FILE='IERRNO')
        WRITE(9,520) IC
        CLOSE(9)
      ENDIF
C-----------------------------------------------------------------------


C-- FORMAT STATEMENTS ------------------------------------------------
  500 FORMAT(I5,2X,4(1P,E12.5,2X))
  510 FORMAT(' INPUT IRCODE: 1 -- END RUN, ELSE -- CONTINUE'/
     A       ' RE =',F7.0,4X,'SIGZET =',F8.2)
  520 FORMAT(' ',I5)
C-----------------------------------------------------------------------

      END
```

```
      SUBROUTINE ROTZET(ISWTCH)
*     PARAMETER ( M = 101, N = M )
      INCLUDE (PARM)
*****************************************************************************
*                                                                           *
*   SOLVE VORTICITY FOR ROTATING FLOW USING ADI AND STRETCHING              *
*   TRANSFORMATION                                                          *
*                                                                           *
*****************************************************************************
      INTEGER ISWTCH, NETA, NXI
      DOUBLE PRECISION AR, B, BETA, DETA, DXI, D1ETAZ, D1XIR, D2ETAZ,
     A  D2XIR, PSI, R, RE, SIGZET, U, V, W, Z, ZETA
      DOUBLE PRECISION A(N), AA(M), AAT(M), AC(N), ACA(M), AT(N),
     A  BN(M), BP(M), C(N), CA(M), CAT(M), CT(N), DN(N`  DP(N),
     A  F(N), FA(M), WK1(M), WK2(N), ZETAA(M,N)

      COMMON /BLOCKG/ DXI, DETA, R(M), Z(N), D1XIR(M), D2XIR(M),
     A  D1ETAZ(N), D2ETAZ(N)
      COMMON /BLOCKN/ NXI, NETA
      COMMON /BLOCKP/ RE, AR, B, SIGZET
      COMMON /BLOCKV/ PSI(M,N,2), U(M,N), V(M,N), W(M,N), ZETA(M,N)

C-- SET INITIAL DATA ONLY THE FIRST TIME THROUGH ----------------------
      IF ( ISWTCH.EQ.1 ) THEN
        ISWTCH = 0
        BETA = DXI/DETA
        DO 10 I = 2,NXI-1
          AA(I) = - D1XIR(I)**2 + DXI/2.*(D2XIR(I)+D1XIR(I)/R(I))
          CA(I) = - D1XIR(I)**2 - DXI/2.*(D2XIR(I)+D1XIR(I)/R(I))
          ACA(I) = DXI*RE/2.*D1XIR(I)
          BP(I) = SIGZET + 2.*D1XIR(I)**2 + (DXI/R(I))**2
   10     BN(I) = SIGZET - 2.*D1XIR(I)**2 - (DXI/R(I))**2
        DO 20 J = 2,NETA-1
          A(J) = - (BETA/AR*D1ETAZ(J))**2 + DXI*BETA/2./AR**2*D2ETAZ(J)
          C(J) = - (BETA/AR*D1ETAZ(J))**2 - DXI*BETA/2./AR**2*D2ETAZ(J)
          AC(J) = DXI*BETA*RE/2.*D1ETAZ(J)
          DP(J) = SIGZET + 2.*(BETA/AR*D1ETAZ(J))**2
   20     DN(J) = SIGZET - 2.*(BETA/AR*D1ETAZ(J))**2
      ENDIF
C-----------------------------------------------------------------------

C-- TRANSFER BOUNDARY ZETA TO TEMPORARY STORAGE ------------------------
      DO 30 I = 1,NXI
        ZETAA(I,1) = ZETA(I,1)
   30   ZETAA(I,NETA) = ZETA(I,NETA)
      DO 40 J = 2,NETA-1
        ZETAA(1,J) = ZETA(1,J)
   40   ZETAA(NXI,J) = ZETA(NXI,J)
C-----------------------------------------------------------------------


C-----------------------------------------------------------------------
C            FIRST TIME STEP -- SOLVE BY LINES
C-----------------------------------------------------------------------

C---- CALCULATE COEFFICIENTS AND RIGHT-HAND SIDE ----------------------
      DO 70 J = 2,NETA-1
        DO 50 I = 2,NXI-1
          AAT(I) = - ACA(I)*U(I-1,J) + AA(I)
          CAT(I) =   ACA(I)*U(I+1,J) + CA(I)
```

```
   50     FA(I) = (   AC(J)*W(I,J-1) - A(J) )*ZETA(I,J-1)
     A       +     ( - AC(J)*W(I,J+1) - C(J) )*ZETA(I,J+1)
     A       + DN(J)*ZETA(I,J)
     A       + 2.*AC(J)/R(I)*( V(I,J+1) - V(I,J-1) )*V(I,J)
          FA(2)      = FA(2)      - AAT(2)*ZETA(1,J)
          FA(NXI-1) = FA(NXI-1) - CAT(NXI-1)*ZETA(NXI,J)
C-------------------------------------------------------------------------

C---- TRIDIAGONAL SOLVER ------------------------------------------------
        CALL TRIDBL(AAT,BP,CAT,FA,WK1,NXI,2,NXI-1)
C-------------------------------------------------------------------------

C---- TRANSFER SOLUTION TO TEMPORARY STORAGE-----------------------------
        DO 60 I = 2,NXI-1
   60     ZETAA(I,J) = FA(I)
   70 CONTINUE
C-------------------------------------------------------------------------


C-------------------------------------------------------------------------
C            SECOND TIME STEP -- SOLVE BY LINES
C-------------------------------------------------------------------------

C---- CALCULATE COEFFICIENTS AND RIGHT-HAND SIDE ------------------------
      DO 100 I = 2,NXI-1
        DO 80 J = 2,NETA-1
          AT(J) = - AC(J)*W(I,J-1) + A(J)
          CT(J) =   AC(J)*W(I,J+1) + C(J)
   80     F(J) = (   ACA(I)*U(I-1,J) - AA(I) )*ZETAA(I-1,J)
     A       +    ( - ACA(I)*U(I+1,J) - CA(I) )*ZETAA(I+1,J)
     A       + BN(I)*ZETAA(I,J)
     A       + 2.*AC(J)/R(I)*( V(I,J+1) - V(I,J-1) )*V(I,J)
          F(2)       = F(2)       - AT(2)*ZETA(I,1)
          F(NETA-1) = F(NETA-1) - CT(NETA-1)*ZETA(I,NETA)
C-------------------------------------------------------------------------

C---- TRIDIAGONAL SOLVER ------------------------------------------------
        CALL TRIDBL(AT,DP,CT,F,WK2,NETA,2,NETA-1)
C-------------------------------------------------------------------------

C---- TRANSFER SOLUTION -------------------------------------------------
        DO 90 J = 2,NETA-1
   90     ZETA(I,J) = F(J)
  100 CONTINUE
C-------------------------------------------------------------------------

      END
```

```
      SUBROUTINE ROTPSI(ISWTCH)
*     PARAMETER ( M = 101, N = M )
      INCLUDE (PARM)
************************************************************************
*                                                                      *
*   SOLVE STEAM-FUNCTION FOR ROTATING FLOW USING ADI AND               *
*   STRETCHING TRANSFORMATION                                          *
*                                                                      *
************************************************************************
      INTEGER ISWTCH, NETA, NXI
      DOUBLE PRECISION AR, B, BETA, DETA, DXI, D1ETAZ, D1XIR, D2ETAZ,
     A  D2XIR, PSI, R, RE, SIGPSI, SIGZET, U, V, W, Z, ZETA
      DOUBLE PRECISION A(N), AA(M), BN(M), BP(M), C(N), CA(M), DN(N),
     A  DP(N), F(N), FA(M), WK1(M), WK2(N), PSIA(M,N)

      COMMON /BLOCKG/ DXI, DETA, R(M), Z(N), D1XIR(M), D2XIR(M),
     A  D1ETAZ(N), D2ETAZ(N)
      COMMON /BLOCKN/ NXI, NETA
      COMMON /BLOCKP/ RE, AR, B, SIGZET
      COMMON /BLOCKV/ PSI(M,N,2), U(M,N), V(M,N), W(M,N), ZETA(M,N)

C-- SET INITIAL DATA ONLY THE FIRST TIME THROUGH ----------------------
      IF ( ISWTCH.EQ.1 ) THEN
        ISWTCH = 0
        BETA = DXI/DETA
        SIGPSI = SIGZET/RE
        DO 10 I = 2,NXI-1
          AA(I) =-(AR*D1XIR(I))**2+DXI*AR**2/2.*(D2XIR(I)-D1XIR(I)/R(I))
          CA(I) =-(AR*D1XIR(I))**2-DXI*AR**2/2.*(D2XIR(I)-D1XIR(I)/R(I))
          BP(I) = SIGPSI + 2.*(AR*D1XIR(I))**2
   10     BN(I) = SIGPSI - 2.*(AR*D1XIR(I))**2
        DO 20 J = 2,NETA-1
          A(J)  = - (BETA*D1ETAZ(J))**2 + DXI*BETA/2.*D2ETAZ(J)
          C(J)  = - (BETA*D1ETAZ(J))**2 - DXI*BETA/2.*D2ETAZ(J)
          DP(J) = SIGPSI + 2.*(BETA*D1ETAZ(J))**2
   20     DN(J) = SIGPSI - 2.*(BETA*D1ETAZ(J))**2

C---- TRANSFER BOUNDARY PSI TO TEMPORARY STORAGE ----------------------
        DO 30 I = 1,NXI
          PSIA(I,1)    = PSI(I,1,1)
   30     PSIA(I,NETA) = PSI(I,NETA,1)
        DO 40 J = 1,NETA
          PSIA(1,J)    = PSI(1,J,1)
   40     PSIA(NXI,J) = PSI(NXI,J,1)
      ENDIF
C----------------------------------------------------------------------


C----------------------------------------------------------------------
C             FIRST TIME STEP -- SOLVE BY LINES
C----------------------------------------------------------------------

C---- CALCULATE COEFFICIENTS AND RIGHT-HAND SIDE ----------------------
      DO 70 J = 2,NETA-1
        DO 50 I = 2,NXI-1
   50     FA(I) = - A(J)*PSI(I,J-1,1) + DN(J)*PSI(I,J,1)
     A           - C(J)*PSI(I,J+1,1) - DXI**2*R(I)*ZETA(I,J)
        FA(2)     = FA(2)     - AA(2)*PSI(1,J,1)
        FA(NXI-1) = FA(NXI-1) - CA(NXI-1)*PSI(NXI,J,1)
C----------------------------------------------------------------------
```

```
C---- TRIDIAGONAL SOLVER ------------------------------------------------
        CALL TRIDBL(AA,BP,CA,FA,WK1,NXI,2,NXI-1)
C-----------------------------------------------------------------------

C---- TRANSFER SOLUTION TO TEMPORARY STORAGE ----------------------------
        DO 60 I = 2,NXI-1
   60     PSIA(I,J) = FA(I)
   70 CONTINUE
C-----------------------------------------------------------------------


C-----------------------------------------------------------------------
C             SECOND TIME STEP -- SOLVE BY LINES
C-----------------------------------------------------------------------

C---- CALCULATE COEFFICIENTS AND RIGHT-HAND SIDE ------------------------
      DO 100 I = 2,NXI-1
        DO 80 J = 2,NETA-1
   80     F(J) = - AA(I)*PSIA(I-1,J) + BN(I)*PSIA(I,J)
     A           - CA(I)*PSIA(I+1,J) - DXI**2*R(I)*ZETA(I,J)
        F(2)       = F(2)       - A(2)*PSI(I,1,1)
        F(NETA-1) = F(NETA-1) - C(NETA-1)*PSI(I,NETA,1)
C-----------------------------------------------------------------------

C---- TRIDIAGONAL SOLVER ------------------------------------------------
        CALL TRIDBL(A,DP,C,F,WK2,NETA,2,NETA-1)
C-----------------------------------------------------------------------

C---- TRANSFER SOLUTION -------------------------------------------------
        DO 90 J = 2,NETA-1
   90     PSI(I,J,2) = F(J)
  100 CONTINUE
C-----------------------------------------------------------------------

      END
```

```
      SUBROUTINE ROTVEL(ISWTCH)
*     PARAMETER ( M = 101, N = M )
      INCLUDE (PARM)
************************************************************************
*                                                                      *
*   SOLVE V-VELOCITY FOR ROTATING FLOW USING ADI AND STRETCHING        *
*   TRANSFORMATION                                                     *
*                                                                      *
************************************************************************
      INTEGER ISWTCH, NETA, NXI
      DOUBLE PRECISION AR, B, BETA, DETA, DXI, D1ETAZ, D1XIR, D2ETAZ,
     A  D2XIR, PSI, R, RE, SIGVEL, SIGZET, U, V, W, Z, ZETA
      DOUBLE PRECISION A(N), AA(M), AAT(M), AC(N), ACA(M), AT(N), BA(M),
     A  BAT(M), BN(M), BP(M), C(N), CA(M), CAT(M), CT(N), DN(N), DP(N),
     A  F(N), FA(M), WK1(M), WK2(N), VA(M,N)

      COMMON /BLOCKG/ DXI, DETA, R(M), Z(N), D1XIR(M), D2XIR(M),
     A  D1ETAZ(N), D2ETAZ(N)
      COMMON /BLOCKN/ NXI, NETA
      COMMON /BLOCKP/ RE, AR, B, SIGZET
      COMMON /BLOCKV/ PSI(M,N,2), U(M,N), V(M,N), W(M,N), ZETA(M,N)

C-- SET INITIAL DATA ONLY THE FIRST TIME THROUGH ----------------------
      IF ( ISWTCH.EQ.1 ) THEN
        ISWTCH = 0
        BETA = DXI/DETA
        SIGVEL = SIGZET/10.
        DO 10 I = 2,NXI-1
          AA(I) = - D1XIR(I)**2 + DXI/2.*(D2XIR(I)+D1XIR(I)/R(I))
          CA(I) = - D1XIR(I)**2 - DXI/2.*(D2XIR(I)+D1XIR(I)/R(I))
          ACA(I) = DXI*RE/2.*D1XIR(I)
          BA(I) = 2.*DXI**2*RE/R(I)
          BP(I) = SIGVEL + 2.*D1XIR(I)**2 + (DXI/R(I))**2
   10     BN(I) = SIGVEL - 2.*D1XIR(I)**2 - (DXI/R(I))**2
        DO 20 J = 2,NETA-1
          A(J) = - (BETA/AR*D1ETAZ(J))**2 + DXI*BETA/2./AR**2*D2ETAZ(J)
          C(J) = - (BETA/AR*D1ETAZ(J))**2 - DXI*BETA/2./AR**2*D2ETAZ(J)
          AC(J) = DXI*BETA*RE/2.*D1ETAZ(J)
          DP(J) = SIGVEL + 2.*(BETA/AR*D1ETAZ(J))**2
   20     DN(J) = SIGVEL - 2.*(BETA/AR*D1ETAZ(J))**2

C---- TRANSFER BOUNDARY V TO TEMPORARY STORAGE ONLY THE FIRST TIME -----
        DO 30 I = 1,NXI
          VA(I,1) = V(I,1)
   30     VA(I,NETA) = V(I,NETA)
        DO 40 J = 1,NETA
          VA(1,J) = V(1,J)
   40     VA(NXI,J) = V(NXI,J)
C----------------------------------------------------------------------
      ENDIF
C----------------------------------------------------------------------


C----------------------------------------------------------------------
C            FIRST TIME STEP -- SOLVE BY LINES
C----------------------------------------------------------------------

C---- CALCULATE COEFFICIENTS AND RIGHT-HAND SIDE ----------------------
      DO 70 J = 2,NETA-1
        DO 50 I = 2,NXI-1
```

```
          AAT(I) = - ACA(I)*U(I-1,J) + AA(I)
          BAT(I) =   BA(I)*U(I,J) + BP(I)
          CAT(I) =   ACA(I)*U(I+1,J) + CA(I)
   50     FA(I) = (   AC(J)*W(I,J-1) - A(J) )*V(I,J-1)
     A      +     ( - AC(J)*W(I,J+1) - C(J) )*V(I,J+1)
     A      + DN(J)*V(I,J)
          FA(2)      = FA(2)      - AAT(2)*V(1,J)
          FA(NXI-1) = FA(NXI-1) - CAT(NXI-1)*V(NXI,J)
C-------------------------------------------------------------------

C---- TRIDIAGONAL SOLVER ---------------------------------------------
        CALL TRIDBL(AAT,BAT,CAT,FA,WK1,NXI,2,NXI-1)
C-------------------------------------------------------------------

C---- TRANSFER SOLUTION TO TEMPORARY STORAGE-------------------------
        DO 60 I = 2,NXI-1
   60     VA(I,J) = FA(I)
   70 CONTINUE
C-------------------------------------------------------------------


C-------------------------------------------------------------------
C             SECOND TIME STEP -- SOLVE BY LINES
C-------------------------------------------------------------------

C---- CALCULATE COEFFICIENTS AND RIGHT-HAND SIDE ----------------------
      DO 100 I = 2,NXI-1
        DO 80 J = 2,NETA-1
          AT(J) = - AC(J)*W(I,J-1) + A(J)
          CT(J) =   AC(J)*W(I,J+1) + C(J)
   80     F(J) = (   ACA(I)*U(I-1,J) - AA(I) )*VA(I-1,J)
     A      +     ( - ACA(I)*U(I+1,J) - CA(I) )*VA(I+1,J)
     A      + ( - BA(I)*U(I,J) + BN(I) )*VA(I,J)
          F(2)      = F(2)      - AT(2)*V(I,1)
          F(NETA-1) = F(NETA-1) - CT(NETA-1)*V(I,NETA)
C-------------------------------------------------------------------

C---- TRIDIAGONAL SOLVER ---------------------------------------------
        CALL TRIDBL(AT,DP,CT,F,WK2,NETA,2,NETA-1)
C-------------------------------------------------------------------

C---- TRANSFER SOLUTION ----------------------------------------------
        DO 90 J = 2,NETA-1
   90     V(I,J) = F(J)
  100 CONTINUE
C-------------------------------------------------------------------

      END
```

```
      SUBROUTINE TRIDBL(A,B,C,F,X,N,NT,NB)
*********************************************************************
*                                                                   *
*    TRIDBL:  TRIDIAGONAL SOLVER USING DOUBLE PRECISION             *
*                                                                   *
*    KEY:                                                           *
*      A(*) = LOWER DIAGONAL                                        *
*      B(*) = DIAGONAL                                              *
*      C(*) = UPPER DIAGONAL                                        *
*      X(*) = WORKING ARRAY                                         *
*      F(*) = F ARRAY (CONTAINS SOLUTION AT EXIT FROM MODULE)       *
*      N = LENGTH OF ARRAY                                          *
*      NB = BOTTOM LIMIT FOR THE CALCULATION                        *
*      NT = TOP LIMIT FOR THE CALCULATION                           *
*                                                                   *
*********************************************************************
      INTEGER N,NB,NT
      DOUBLE PRECISION A(N),B(N),C(N),F(N),X(N),Z

C--- NORMALIZE FIRST ROW ----------------------------------------------
      X(NT) = C(NT)/B(NT)
      F(NT) = F(NT)/B(NT)
C---------------------------------------------------------------------

C--- REDUCE THE DIAGONAL TO 1'S ---------------------------------------
      DO 10 J = NT+1,NB
         Z = 1./(B(J) - A(J)*X(J-1))
         X(J) = C(J)*Z
         F(J) = (F(J) - A(J)*F(J-1))*Z
   10 CONTINUE
C---------------------------------------------------------------------

C--- SWEEP BACKWARDS TO SOLVE PLACING THE SOLUTION IN F ---------------
      DO 20 J = NB-1,NT,-1
         F(J) = F(J) - X(J)*F(J+1)
   20 CONTINUE
C---------------------------------------------------------------------

      END
```

```
      SUBROUTINE MAXABS(A,M,N,MX,NY,AMAX,IMAX,JMAX)
*************************************************************************
*                                                                       *
*  FIND THE MAXIMUM ABSOLUTE VALUE OF AND ARRAY                         *
*                                                                       *
*       A = MATRIX OF FUNCTION VALUES                                   *
*       AMAX = MAXIMUM VALUE                                            *
*       IMAX = X-LOCATION OF MAXIMUM                                    *
*       JMAX = Y-LOCATION OF MAXIMUM                                    *
*       M = DIMENSION OF A IN X-DIRECTION                               *
*       N = DIMENSION OF A IN Y-DIRECTION                               *
*       MX = NUMBER OF POINTS IN X-DIRECTION TO INTEGRATE               *
*       NY = NUMBER OF POINTS IN Y-DIRECTION TO INTEGRATE               *
*                                                                       *
*************************************************************************
      INTEGER IMAX, JMAX, M, MX, N, NY
      DOUBLE PRECISION A(M,N), AMAX

      IMAX = 0
      JMAX = 0
      AMAX = 0.
      DO 10 I = 1,MX
        DO 10 J = 1,NY
          IF(DABS(A(I,J)).GT.DABS(AMAX))THEN
            AMAX = A(I,J)
            IMAX = I
            JMAX = J
          ENDIF
   10 CONTINUE

      END

      SUBROUTINE D2INTE(A,M,N,MX,NY,DX,DY,AI)
*************************************************************************
*                                                                       *
*  INTEGRATE A FUNCTION IN A PLANE USING EQ. 3-533B OF ROACHE.          *
*                                                                       *
*       A = MATRIX OF FUNCTION VALUES                                   *
*       AI = VALUE OF THE INTEGRAL                                      *
*       DX = STEP SIZE IN X-DIRECTION                                   *
*       DY = STEP SIZE IN Y-DIRECTION                                   *
*       M = DIMENSION OF A IN X-DIRECTION                               *
*       N = DIMENSION OF A IN Y-DIRECTION                               *
*       MX = NUMBER OF POINTS IN X-DIRECTION TO INTEGRATE               *
*       NY = NUMBER OF POINTS IN Y-DIRECTION TO INTEGRATE               *
*                                                                       *
*************************************************************************
      INTEGER M, MX, N, NY
      DOUBLE PRECISION AI, DX, DY, A(M,N)


C-- INTEGRATE THE FUNCTION ----------------------------------------------
      AI = (A(1,1) + A(1,NY) + A(MX,1) + A(MX,NY))/4.
      DO 10 I = 2,MX-1
        DO 10 J = 2,NY-1
          AI = AI + A(I,J)
10      CONTINUE
      DO 20 I = 2,MX-1
20      AI = AI + (A(I,1) + A(I,NY))/2.
      DO 30 J = 2,NY-1
```

```
   30     AI = AI + (A(1,J) + A(MX,J))/2.
        AI = AI*DX*DY
C-----------------------------------------------------------------------

      END

      SUBROUTINE SIMSON(A,N,DX,AI)
************************************************************************
*                                                                      *
*  INTEGRATE A FUNCTION USING SIMPSON'S RULE.  IF THE NUMBER OF POINTS *
*    IS EVEN THEN USE THE TRAPEZOID RULE FOR THE FIRST INTERVAL.       *
*                                                                      *
*       A = MATRIX OF FUNCTION VALUES                                  *
*       AI = VALUE OF THE INTEGRAL                                     *
*       DX = STEP SIZE                                                 *
*       N = NUMBER OF DATA POINTS                                      *
*                                                                      *
************************************************************************
      INTEGER IFLAG, N
      DOUBLE PRECISION AI, DX, A(N)

C-- CHECK IF THE NUMBER OF POINTS IS EVEN OR ODD ----------------------
      IFLAG = 0
      IF((N/2)*2.NE.N) IFLAG = 1
C-----------------------------------------------------------------------

C-- INTEGRATE THE FUNCTION ---------------------------------------------
      IF(IFLAG.EQ.1)THEN
        AI = A(1) + 4.*A(2) + A(N)
        DO 10 I = 3,N-2,2
   10     AI = AI + 2.*A(I) + 4.*A(I+1)
        AI = AI*DX/3.
      ELSE
        AI = 3.*(A(1) + A(2))/2. + A(2) + 4.*A(3) + A(N)
        DO 20 I = 4,N-2,2
   20     AI = AI + 2.*A(I) + 4.*A(I+1)
        AI = AI*DX/3.
      ENDIF
C-----------------------------------------------------------------------

      END
```

```
      SUBROUTINE INRUPT(ICODE,IERRNO)
*************************************************************************
*                                                                       *
*  USER INTERRUPT                                                       *
*                                                                       *
*      IERRNO = 207 -- OVERFLOW                                         *
*               208 -- UNDERFLOW                                        *
*               209 -- DIVIDE CHECK                                     *
*                                                                       *
*************************************************************************
      INTEGER ICODE, IERRNO

      OPEN(9,FILE='IERRNO')
      WRITE(6,500)IERRNO
      WRITE(9,510)IERRNO
      WRITE(13,500)IERRNO
      STOP

C-- FORMAT STATEMENTS ------------------------------------------------
  500 FORMAT(' ','INTERRUPT ERROR.  RUN TERMINATED.'/' IERRNO =',I4)
  510 FORMAT(' ',I4)
C-----------------------------------------------------------------------

      END

      SUBROUTINE ERRNU2
*************************************************************************
*                                                                       *
*   DOES NOT INCLUDE OVERFLOW, UNDERFLOW OR DIVIDE CHECK                *
*                                                                       *
*************************************************************************

C-- SET ALLOWABLE NUMBER OF ERRORS TO ONE ----------------------------
      CALL ERRSET(152,1,0,0,0,152)
      CALL ERRSET(154,1,0,0,0,155)
      CALL ERRSET(159,1,0,0,0,161)
      CALL ERRSET(166,1,0,0,0,166)
      CALL ERRSET(169,1,0,0,0,189)
      CALL ERRSET(191,1,0,0,0,197)
      CALL ERRSET(199,1,0,0,0,201)
      CALL ERRSET(203,1,0,0,0,204)
      CALL ERRSET(206,1,0,0,0,206)
      CALL ERRSET(210,1,0,0,0,229)
      CALL ERRSET(231,1,0,0,0,239)
      CALL ERRSET(241,1,0,0,0,301)
C-----------------------------------------------------------------------

      END
```

APPENDIX D: DATA FOR OPTIMIZATION STUDY

File: OPT.WQ1

Re = 10

Grid	Dt(opt)	W(opt)	I(opt)
21x21	0.019	1.72	66
36x36	0.0075	1.81	123
51x51	0.0042	1.83	203
66x66	0.0026	1.84	299
81x81	0.0018	1.84	410
101x101	0.0011	1.83	604

Re = 100

Grid	Dt(opt)	W(opt)	I(opt)
21x21	0.18	1.66	67
36x36	0.077	1.74	146
51x51	0.042	1.77	246
66x66	0.027	1.79	359
81x81	0.018	1.8	488
101x101	0.012	1.81	682

Re = 250

Grid	Dt(opt)	W(opt)	I(opt)
21x21			
36x36	0.192	1.75	226
51x51	0.09	1.78	394
66x66	0.053	1.81	590
81x81	0.035	1.84	807
101x101	0.024	1.84	1111

Re = 500

Grid	Dt(opt)	W(opt)	I(opt)
21x21	0.355	1.03	166
36x36	0.19	1.85	360
51x51	0.155	1.85	467
66x66	0.081	1.9	697
81x81	0.048	1.95	886
101x101	0.042	1.81	1526

Re = 750

Grid	Dt(opt)	W(opt)	I(opt)
21x21			
36x36	0.171	1.91	553
51x51	0.123	1.91	687
66x66	0.105	1.9	844
81x81	0.064	1.95	1096
101x101	0.063	1.8	1738

Re = 1000

Grid	Dt(opt)	W(opt)	I(opt)
21x21			
36x36	0.106	1.95	1211
51x51	0.114	1.95	882
66x66	0.09	1.95	1073
81x81	0.076	1.95	1268
101x101	0.085	1.79	2025

APPENDIX E: DATA FOR ROTATING FLOW

*--indicates nonuniform grid was used

File: A0P02.WQ1
Rotating Flow--Bulk Data
$\delta = 0.02$

Series:RTAA
b = 0.10

Re	Cm	Qp	Qs	Grid
1	206.8	0.004488	9.95e-08	501x11
10	20.68	0.004488	0.000001	501x11
100	2.068	0.004488	0.00001	501x11
500	0.4137	0.004488	0.000051	501x11
1000	0.2069	0.004488	0.000102	501x11
5000	0.04209	0.004481	0.000512	501x11
10000	0.02216	0.00446	0.00102	501x11

Series:RTAB
b = 0.05

Re	Cm	Qp	Qs	Grid
1	257.4	0.004741	1.10e-07	501x11
10	25.74	0.004741	0.000001	501x11
100	2.574	0.004741	0.000011	501x11
500	0.5148	0.004741	0.000057	501x11
1000	0.2575	0.004741	0.000113	501x11
5000	0.05231	0.004733	0.000568	501x11
10000	0.0275	0.00471	0.001136	501x11

Series:RTAC
b = 0.025

Re	Cm	Qp	Qs	Grid
1	287	0.004861	1.17e-07	601x13
10	28.7	0.004861	0.000001	601x13
100	2.87	0.004861	0.000012	601x13
500	0.5741	0.004861	0.00006	601x13
1000	0.2872	0.004861	0.00012	601x13
5000	0.0582	0.004852	0.000599	601x13
10000	0.03045	0.004826	0.001204	601x13

Series:RTAD
b = 0.01

Re	Cm	Qp	Qs	Grid
1	309.5	0.004924	1.22e-07	601x13
10	30.95	0.004924	0.000001	601x13
100	3.095	0.004924	0.000012	601x13
500	0.6191	0.004923	0.000061	601x13
1000	0.3096	0.004923	0.000122	601x13
5000	0.06254	0.004913	0.000613	601x13
10000	0.03244	0.004882	0.001238	601x13

Series:RTAE
b = 0.002

Re	Cm	Qp	Qs	Grid
1	334.5	0.004935	1.17e-07	601x13*
10	33.45	0.004935	0.000001	601x13*
100	3.345	0.004935	0.000012	601x13*
500	0.6691	0.004935	0.00006	601x13*
1000	0.3346	0.004934	0.000119	601x13*
5000	0.06734	0.004924	0.000599	601x13*
10000	0.03451	0.004891	0.00121	601x13*

Series:RTAF
b = 0.001

Re	Cm	Qp	Qs	Grid
1	339.6	0.004939	1.20e-07	801x17*
10	33.96	0.004939	0.000001	801x17*
100	3.3396	0.004939	0.000012	801x17*
500	0.6792	0.004939	0.00006	801x17*
1000	0.3397	0.004939	0.000121	801x17*
5000	0.06831	0.004928	0.000607	801x17*
10000	0.03492	0.004895	0.001225	801x17*

File: A0P10.WQ1
Rotating Flow--Bulk Data
$\delta = 0.10$

Series:RTBA
b = 0.10

Re	Cm	Qp	Qs	Grid
1	44.57	0.02204	0.000011	161x17
10	4.458	0.02204	0.00011	161x17
100	0.4469	0.02203	0.001101	161x17
500	0.09518	0.0218	0.005487	161x17
1000	0.05709	0.0212	0.01067	161x17
5000	0.02902	0.0192	0.02004	321x33
10000	0.02112	0.01989	0.01603	321x33

Series:RTBB
b = 0.05

Re	Cm	Qp	Qs	Grid
1	58.28	0.02306	0.000012	161x17
10	5.828	0.02306	0.000119	161x17
100	0.5839	0.02305	0.001188	161x17
500	0.1225	0.02278	0.005945	161x17
1000	0.07118	0.02207	0.01159	161x17
5000	0.03518	0.01957	0.02107	321x33
10000	0.02567	0.02036	0.01595	321x33

Series:RTBC
b = 0.025

Re	Cm	Qp	Qs	Grid
1	69.89	0.02344	0.000012	161x17
10	6.989	0.02344	0.000122	161x17
100	0.6998	0.02343	0.00122	161x17
500	0.1448	0.02315	0.006083	161x17
1000	0.08133	0.0224	0.01183	161x17
5000	0.03798	0.01975	0.02114	321x33
10000	0.02775	0.02058	0.01551	321x33

Series:RTBDX
b = 0.01

Re	Cm	Qp	Qs	Grid
1	84.52	0.02359	0.000012	301x31
10	8.452	0.02358	0.000123	301x31
100	0.846	0.02357	0.001233	301x31
500	0.1732	0.02329	0.006146	301x31
1000	0.09422	0.02253	0.01192	301x31
5000	0.04016	0.01983	0.02112	301x31
10000	0.02898	0.02066	0.01549	301x31

File: A0P25.WQ1
Rotating Flow--Bulk Data
$\delta = 0.25$

Series:RTCA
b = 0.10

Re	Cm	Qp	Qs	Grid
1	24.09	0.05112	0.000126	161x41
10	2.41	0.05111	0.00126	161x41
100	0.2509	0.05033	0.01243	161x41
500	0.08928	0.04285	0.04416	161x41
1000	0.06825	0.04137	0.05078	161x41
5000	0.03175	0.04266	0.02781	210x51
10000	0.02306	0.04309	0.02108	210x51

Series:RTCB
b = 0.05

Re	Cm	Qp	Qs	Grid
1	34.73	0.05286	0.000132	161x41
10	3.474	0.05285	0.001316	161x41
100	0.3566	0.05202	0.01298	161x41
500	0.1104	0.04418	0.04532	161x41
1000	0.08144	0.04274	0.05121	161x41
5000	0.03827	0.04432	0.02683	201x51
10000	0.02794	0.04493	0.01988	201x51

Series:RTCC
b = 0.025

Re	Cm	Qp	Qs	Grid
1	45.37	0.05343	0.000133	161x41
10	4.538	0.05342	0.00133	161x41
100	0.4623	0.05258	0.01311	161x41
500	0.1298	0.04465	0.04555	161x41
1000	0.09066	0.04322	0.05122	161x41
5000	0.04122	0.04481	0.02644	201x51
10000	0.03024	0.04546	0.01956	201x51

Series:RTCD
b = 0.01

Re	Cm	Qp	Qs	Grid
1	59.31	0.05363	0.000133	201x51
10	5.932	0.05362	0.001335	201x51
100	0.6014	0.05278	0.01315	201x51
500	0.1563	0.04483	0.04561	201x51
1000	0.1029	0.04337	0.05114	201x51
5000	0.04324	0.04497	0.02626	201x51
10000	0.03131	0.0456	0.01946	201x51

File: A0P50.WQ1
Rotating Flow--Bulk Data
$\delta = 0.50$

Series:RTDA
b = 0.10

Re	Cm	Qp	Qs	Grid
1	20.15	0.08513	0.000558	161x81
10	2.019	0.085	0.005571	161x81
100	0.2342	0.07674	0.04677	161x81
500	0.09962	0.069	0.07523	161x81
1000	0.0713	0.07238	0.05834	161x81
5000	0.03318	0.07244	0.03135	201x101
10000	0.02403	0.07401	0.02386	201x101

Series:RTDB
b = 0.05

Re	Cm	Qp	Qs	Grid
1	30.42	0.08749	0.000574	161x81
10	3.045	0.08735	0.005725	161x81
100	0.3359	0.07882	0.04787	161x81
500	0.1213	0.07124	0.07536	161x81
1000	0.0852	0.0744	0.05766	161x81
5000	0.04016	0.07553	0.03021	201x101
10000	0.02929	0.07762	0.02277	201x101

Series:RTDC
b = 0.025

Re	Cm	Qp	Qs	Grid
1	40.95	0.08822	0.000578	161x81
10	4.099	0.08808	0.005764	161x81
100	0.4404	0.07946	0.04795	161x81
500	0.1408	0.07193	0.07543	161x81
1000	0.09459	0.07511	0.05745	161x81
5000	0.04328	0.07641	0.0299	201x101
10000	0.03172	0.07869	0.02245	201x101

Series:RTDD
b = 0.01

Re	Cm	Qp	Qs	Grid
1	54.87	0.08846	0.000578	201x101
10	5.49	0.08832	0.005767	201x101
100	0.5791	0.0797	0.0482	201x101
500	0.1673	0.07218	0.0753	201x101
1000	0.1069	0.07519	0.05729	201x101
5000	0.04535	0.07666	0.02982	201x101
10000	0.03282	0.07891	0.02236	201x101

File: A0P75.WQ1
Rotating Flow--Bulk Data
$\delta = 0.75$

Series:RTEA
b = 0.10

Re	Cm	Qp	Qs	Grid
1	19.66	0.1027	0.000985	121x91
10	1.972	0.1024	0.009788	121x91
100	0.2411	0.08732	0.06642	121x91
500	0.1023	0.08938	0.07623	121x91
1000	0.07398	0.08946	0.06006	121x91
5000	0.03532	0.09767	0.03339	121x91
10000	0.02479	0.1027	0.02613	121x91

Series:RTEB
b = 0.05

Re	Cm	Qp	Qs	Grid
1	29.86	0.1054	0.001008	121x91
10	2.991	0.105	0.01002	121x91
100	0.3423	0.08961	0.0676	121x91
500	0.1243	0.09192	0.07608	121x91
1000	0.08825	0.092	0.05974	121x91
5000	0.04293	0.1019	0.0327	121x91
10000	0.03043	0.1077	0.02495	121x91

Series:RTEC
b = 0.025

Re	Cm	Qp	Qs	Grid
1	40.28	0.1062	0.001012	121x91
10	4.033	0.1058	0.01008	121x91
100	0.4457	0.09033	0.06789	121x91
500	0.1436	0.09269	0.07604	121x91
1000	0.09763	0.09276	0.05959	121x91
5000	0.04622	0.1029	0.03249	121x91
10000	0.03331	0.1092	0.02474	121x91

File: A1P00.WQ1
Rotating Flow--Bulk Data
$\delta = 1.00$

Series:RTFA
b = 0.10

Re	Cm	Qp	Qs	Grid
1	19.57	0.1104	0.001192	101x101
10	1.964	0.1099	0.01181	101x101
100	0.2437	0.09238	0.0722	101x101
500	0.1043	0.1014	0.07594	101x101
1000	0.07619	0.1029	0.06173	101x101
5000	0.03599	0.1167	0.03391	121x121
10000	0.02524	0.1244	0.02676	121x121

Series:RTFB
b = 0.05

Re	Cm	Qp	Qs	Grid
1	29.73	0.1132	0.00122	101x101
10	2.979	0.1127	0.01209	101x101
100	0.3446	0.0948	0.07339	101x101
500	0.1264	0.1042	0.0759	101x101
1000	0.09068	0.1059	0.06133	101x101
5000	0.04378	0.1218	0.03404	121x121
10000	0.03104	0.1306	0.0263	121x121

Series:RTFC
b = 0.025

Re	Cm	Qp	Qs	Grid
1	40.21	0.1141	0.001227	121x121
10	4.027	0.1135	0.01215	121x121
100	0.4484	0.09554	0.07362	121x121
500	0.1453	0.1049	0.07574	121x121
1000	0.09937	0.1065	0.06097	121x121
5000	0.04714	0.1231	0.03397	121x121
10000	0.03398	0.1324	0.02609	121x121

File: A1P50.WQ1
Rotating Flow--Bulk Data
$\delta = 1.50$

Series:RTGA
b = 0.10

Re	Cm	Qp	Qs	Grid
1	19.46	0.1148	0.001259	61x91
10	1.952	0.1142	0.01246	61x91
100	0.2445	0.09589	0.0738	61x91
500	0.1088	0.1144	0.07741	61x91
1000	0.08063	0.1247	0.06297	61x91
5000	0.03459	0.1552	0.03894	61x91
10000				

Series:RTGB
b = 0.05

Re	Cm	Qp	Qs	Grid
1	29.46	0.1177	0.001289	61x91
10	2.952	0.1171	0.01275	61x91
100	0.3439	0.09841	0.07497	61x91
500	0.1308	0.1178	0.07741	61x91
1000	0.09543	0.1281	0.06254	61x91
5000	0.04261	0.1627	0.03846	61x91
10000				

Series:RTGC
b = 0.025

Re	Cm	Qp	Qs	Grid
1	39.81	0.1185	0.001294	81x121
10	3.988	0.1179	0.0128	81x121
100	0.4459	0.09913	0.07505	81x121
500	0.1482	0.1181	0.0769	81x121
1000	0.1029	0.1275	0.06156	81x121
5000	0.04872	0.1592	0.03649	81x121
10000				

File: A2P00.WQ1
Rotating Flow--Bulk Data
$\delta = 2.0$

Series:RTHA
b = 0.10

Re	Cm	Qp	Qs	Grid
1	19.46	0.1158	0.00126	61x121
10	1.952	0.1142	0.01248	61x121
100	0.2445	0.09657	0.07382	61x121
500	0.1094	0.1171	0.07768	61x121
1000	0.08152	0.1399	0.06255	61x121
2500	0.05287	0.1573	0.04684	61x121

Series:RTHB
b = 0.05

Re	Cm	Qp	Qs	Grid
1	29.46	0.1184	0.00129	61x121
10	2.952	0.1178	0.01276	61x121
100	0.3439	0.09911	0.075	61x121
500	0.1315	0.1207	0.07765	61x121
1000	0.09642	0.1434	0.06217	61x121
2500	0.06401	0.163	0.04713	61x121

Series:RTHC
b = 0.025

Re	Cm	Qp	Qs	Grid
1	39.81	0.1192	0.001295	81x161
10	3.988	0.1186	0.01282	81x161
100	0.4459	0.09983	0.07507	81x161
500	0.1489	0.121	0.07714	81x161
1000	0.1039	0.1428	0.06115	81x161
2500	0.06836	0.1596	0.04615	81x161

File: A2P50.WQ1
Rotating Flow--Bulk Data
$\delta = 2.5$

Series:RTIA
b = 0.10

				Grid
1	19.46	0.1156	0.001261	61x151
10	1.952	0.115	0.01248	61x151
100	0.2445	0.09668	0.07382	61x151
500	0.1094	0.1177	0.07771	61x151
1000	0.08215	0.1483	0.06232	61x151
2500	0.05334	0.1686	0.04665	61x151

Series:RTIB
b = 0.05

Re	Cm	Qp	Qs	Grid
1	29.46	0.1185	0.00129	61x151
10	2.952	0.1179	0.01276	61x151
100	0.3439	0.09922	0.075	61x151
500	0.1316	0.1214	0.07768	61x151
1000	0.09712	0.1527	0.0619	61x151
2500	0.06457	0.1746	0.04715	61x151

Series:RTIC
b = 0.025

Re	Cm	Qp	Qs	Grid
1	39.81	0.1193	0.001295	81x201
10	3.988	0.1187	0.01282	81x201
100	0.4459	0.09994	0.07508	81x201
500	0.149	0.1217	0.07717	81x201
1000	0.1046	0.152	0.06087	81x201
2500	0.06893	0.1708	0.04623	81x201

File: A3P00.WQ1
Rotating Flow--Bulk Data
$\delta = 3.0$

Series:RTJA
b = 0.10

Re	Cm	Qp	Qs	Grid
1	19.46	0.1156	0.001261	61x181
10	1.952	0.115	0.01248	61x181
100	0.2445	0.0967	0.07382	61x181
500	0.1094	0.118	0.07771	61x181
1000	0.0824	0.1495	0.06228	61x181
2500	0.05364	0.1777	0.04643	61x181

Series:RTJB
b = 0.05

Re	Cm	Qp	Qs	Grid
1	29.46	0.1185	0.00129	61x181
10	2.952	0.1179	0.01276	61x181
100	0.3439	0.09924	0.075	61x181
500	0.1316	0.1216	0.07769	61x181
1000	0.09741	0.1542	0.06185	61x181
2500	0.06493	0.1837	0.04696	61x181

Series:RTJC
b = 0.025

Re	Cm	Qp	Qs	Grid
1	39.81	0.1193	0.001295	81x241
10	3.988	0.1187	0.01282	81x241
100	0.4459	0.09996	0.07508	81x241
500	0.149	0.1219	0.07717	81x241
1000	0.1049	0.1532	0.06083	81x241
2500	0.06929	0.1797	0.046	81x241

VITA AUCTORIS

Edward Lang was born in 1954 in Montreal, Quebec. He graduated from Malcolm Campbell High School in 1972 and in 1974 from Vanier College. In 1980 he graduated from McGill University where he obtained a B.Eng. in Mechanical Engineering. In 1984 he obtained an M.Sc. degree in Applied Mathematics from the University of Western Ontario. His is currently a Ph.D. candidate in Mechanical Engineering at the University of Windsor.